# Cardiovascular Disease Detection Using Deep Learning

**Aadit Trivedi[1], Aarya Shah[2], Dr. M. L. Sworna Kokila[3]**

[1,2,3]Department of Computing, Technologies, College of Engineering and Technology, Student of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, 603203 India

**Abstract**

Cardiovascular diseases have surged as the leading global cause of death, necessitating a focus on early detection and continuous monitoring. This study employs deep learning, specifically the MobileNet Architecture, on a public ECG dataset to predict four cardiac abnormalities: abnormal heartbeat, myocardial infarction, history of myocardial infarction, and normal cases. The developed model demonstrates a notable training accuracy of 97.34% and validation accuracy of 91.00%, showcasing its efficacy in disease classification. With the potential to save lives and reduce healthcare costs, this algorithmic approach offers a reliable, time-efficient alternative to manual diagnosis in detecting heart disorders, providing valuable support to medical professionals..

**Keywords:** Cardiovascular Diseases, Deep Learning, Mobilenet Architecture, ECG Dataset, Early Detection, Disease Classification, Healthcare, Medical Professionals.

## Introduction

In recent decades, cardiovascular diseases have emerged as the predominant cause of mortality worldwide, impacting both industrialized and developing nations. The escalating global health crisis posed by cardiovascular illnesses necessitates a proactive approach, emphasizing early detection and continuous clinical monitoring. The quest for effective strategies to curb the fatality rates associated with these diseases has led to an exploration of innovative technologies, with a particular focus on leveraging artificial intelligence for improved diagnostics.

This study addresses the imperative need for advanced diagnostic tools by proposing an algorithmic model for analyzing electrocardiogram (ECG) tracings. The early identification of cardiac abnormalities, such as abnormal heartbeat, myocardial infarction, and a history of myocardial infarction, is crucial in mitigating the lethal nature of cardiovascular diseases. Employing the MobileNet Architecture, this research achieves remarkable training and validation accuracies, standing at 97.34% and 91.00%, respectively. The proposed model not only exhibits impressive accuracy in disease classification but also holds promise in feature extraction for traditional machine learning classifiers.

The deployment of the MobileNet Architecture as a diagnostic tool can streamline the detection of heart disorders, offering a reliable and efficient alternative to manual methods and providing valuable support to healthcare professionals in their mission to save lives.

## I. STATE OF THE ART

### *A: Revolutionizing Cardiovascular Diagnosis: The Rise of Deep Learning and ECG Datasets*

Current advancements in cardiovascular disease detection predominantly revolve around the integration of artificial intelligence (AI) and deep learning techniques. The utilization of large-scale datasets, such as publicly available electrocardiogram (ECG) datasets, has become a cornerstone in developing robust diagnostic models. Deep learning architectures like the MobileNet Architecture are gaining prominence for their ability to efficiently analyze intricate ECG tracings. Researchers are actively exploring the integration of these models to predict various cardiac abnormalities, offering a more nuanced understanding of heart disorders.

### *B: Algorithmic Frontiers in Cardiovascular Health: Unleashing the Potential of Machine Learning Models*

Recent strides in cardiovascular health underscore the paradigm shift toward algorithmic approaches for disease prediction. Machine learning models, particularly deep learning architectures like MobileNet, are at the forefront of revolutionizing early detection. The convergence of innovative algorithms and extensive ECG datasets allows for nuanced insights into abnormal heart conditions, myocardial infarctions, and related cardiac anomalies. These models not only exhibit impressive accuracy in classification but also pave the way for extracting essential features, propelling the field toward more comprehensive and efficient diagnostic tools. The state of the art in cardiovascular disease research is characterized by a dynamic interplay between advanced computational techniques and a growing understanding of cardiac health indicators.

## II. LITERATURE SURVEY

### 1) A comparative study of classification and prediction of Cardio-Vascular Diseases (CVD) using Machine Learning and Deep Learning techniques

AUTHORS: M. Swathy and K. Saruladha

Cardio-Vascular Diseases (CVD) are found to be rampant in the populace leading to fatal death. The statistics of a recent survey reports that the mortality rate is expanding due to obesity, cholesterol, high blood pressure and usage of tobacco among the people. The severity of the disease is piling up due to the above factors. Studying about the variations of these factors and their impact on CVD is the demand of the hour. This necessitates the usage of modern techniques to identify the disease at its outset and to aid a markdown in the mortality rate. Artificial Intelligence and Data Mining domains have a research scope with their enormous techniques that would assist in the prediction of the CVD priory and identify their behavioural patterns in the large volume of data. The results of these predictions will help the clinicians in decision making and early diagnosis, which would reduce the risk of patients becoming fatal. This paper compares and reports the various Classification, Data Mining, Machine Learning, Deep Learning models that are used for prediction of the Cardio-Vascular diseases. The survey is organized as threefold: Classification and Data Mining Techniques for CVD, Machine Learning Models for CVD and Deep Learning Models for CVD prediction. The performance metrics used for reporting the accuracy, the dataset used for prediction and classification, and the tools used for each category of these techniques are also compiled and reported in this survey.

### 2) Improving electrocardiogram-based detection of rare genetic heart disease using transfer learning: An application to phospholamban p.Arg14del mutation carriers

AUTHORS: R. R. Lopes, H. Bleijendaal, L. A. Ramo, T. E. Verstraelen, A. S. Amin, A. A. Wilde, Y. M. Pinto, B.

A. de Mol and H. A. Marquering

The pathogenic mutation p.Arg14del in the gene encoding Phospholamban (PLN) is known to cause cardiomyopathy and leads to increased risk of sudden cardiac death. Automatic tools might improve the detection of patients with this rare disease. Deep learning is currently the state-of-the-art in signal processing but requires large amounts of data to train the algorithms. In situations with relatively small amounts of data, like PLN, transfer learning may improve accuracy. We propose an ECG-based detection of the PLN mutation using transfer learning from a model originally trained for sex identification. The sex identification model was trained with 256,278 ECGs and subsequently finetuned for PLN detection (155 ECGs of patients with PLN) with two control groups: a balanced age/sex matched group and a randomly selected imbalanced population. The data was split in 10 folds and 20% of the training data was used for validation and early stopping. The models were evaluated with the area under the receiver operating characteristic curve (AUROC) of the testing data. We used gradient activation for explanation of the prediction models. The models trained with transfer learning outperformed the models trained from scratch for both the balanced (AUROC 0.87 vs AUROC 0.71) and imbalanced (AUROC 0.0.90 vs AUROC 0.65) population. The proposed approach was able to improve the accuracy of a rare disease detection model by transfer learning information from a non-manual annotated and abundant label with only limited data available.

### 3) Current methods in electrocardiogram characterization

AUTHORS: R. J. Martis, U. R. Acharya and H. Adeli

The Electrocardiogram (ECG) is the P-QRS-T wave depicting the cardiac activity of the heart. The subtle changes in the electric potential patterns of repolarization and depolarization are indicative of the disease afflicting the patient. These clinical time domain features of the ECG waveform can be used in cardiac health diagnosis. Due to the presence of noise and minute morphological parameter values, it is very difficult to identify the ECG classes accurately by the naked eye. Various computer aided cardiac diagnosis (CACD) systems, analysis methods, challenges addressed and the future of cardiovascular disease screening are reviewed in this paper. Methods developed for time domain, frequency transform domain, and time-frequency domain analysis, such as the wavelet transform, cannot by themselves represent the inherent distinguishing features accurately. Hence, nonlinear methods which can capture the small variations in the ECG signal and provide improved accuracy in the presence of noise are discussed in greater detail in this review. A CACD system exploiting these nonlinear features can help clinicians to diagnose cardiovascular disease more accurately.

### 4) Heart disease detection using deep learning methods from imbalanced ECG samples

AUTHORS: A. Rath, D. Mishra, G. Panda and S. C. Satapathy

Heart disease (HD) is a fatal disease which takes the lives of maximum people compared to other diseases across the world. Early and accurate detection of the disease will help to save many valuable lives. The HD can be detected from medical tests, Electrocardiogram (ECG) signal, heart sounds, Computed Tomography (CT) Images etc. Out of all types of detection of HD from ECG signals plays a vital role. In this paper, the ECG samples of the subjects have been considered as the required inputs to the HD detection model. In recent past, many useful articles have been reported for classification of HD using different machine learning (ML) and deep learning (DL) models. It is observed that with imbalanced HD data the detection accuracy is lower. With an objective to achieve better detection of HD, suitable DL and ML models have been identified in this paper and the required classification models have been developed and tested. The Generative Adversarial Network (GAN) model is chosen with an objective to deal with imbalanced data by generating and using additional fake data for detection purpose. Further, an ensemble model using long short-term memory (LSTM) and GAN is developed in this paper which demonstrates higher performance compared to individual DL model used in this paper. The simulation results using standard MIT-BIH show that the proposed GAN-LSTM model provides the highest accuracy, F1-score and area under curve (AUC) of 0.992, 0.987 and 0.984 respectively compared to other models. Similarly, for PTB-ECG dataset the GAN-LSTM model outperforms all other models with accuracy, F1-score and AUC of 0.994, 0.993 and 0.995 respectively. It is observed that out of the five models investigated, the GAN model performs the best whereas the detection potentiality of the NB model is the lowest. Further research work can be carried out by choosing all other different ensemble models and using other different datasets and the performance can be similarly obtained and compared. The proposed best detection methodology can also be applied to other diseases and healthcare problems.

**5) Cardiac arrhythmia detection using deep learning**

AUTHORS: A. Isin and S. Ozdalili

An electrocardiogram (ECG) is an important diagnostic tool for the assessment of cardiac arrhythmias in clinical routine. In this study, a deep learning framework previously trained on a general image data set is transferred to carry out automatic ECG arrhythmia diagnostics by classifying patient ECG's into corresponding cardiac conditions. Transferred deep convolutional neural network (namely AlexNet) is used as a feature extractor and the extracted features are fed into a simple back propagation neural network to carry out the final classification. Three different conditions of ECG waveform are selected from MIT-BIH arrhythmia database to evaluate the proposed framework. Main focus of this study is to implement a simple, reliable and easily applicable deep learning technique for the classification of the selected three different cardiac conditions. Obtained results demonstrated that the transferred deep learning feature extractor cascaded with a conventional back propagation neural network were able to obtain very high performance rates. Highest obtained correct recognition rate is 98.51% while obtaining testing accuracy around 92%. Based on these results, transferred deep learning proved to be an efficient automatic cardiac arrhythmia detection method while eliminating the burden of training a deep convolutional neural network from scratch providing an easily applicable technique.

## III. SYSTEM MODEL DESIGN

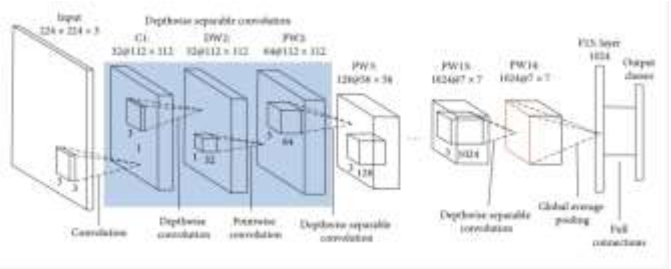
The concept of convolutional neural networks is very successful in image recognition. The key part to understand, which distinguishes CNN from traditional neural networks, is the convolution operation. Having an image at the input, CNN scans it many times to look for certain features. This scanning (convolution) can be set with 2 main parameters: stride and padding type. As we see on below picture, process of the first convolution gives us a set of new frames, shown here in the second column (layer). Each frame contains information about one feature and its presence in scanned image. Resulting frame will have larger values in places where a feature is strongly visible and lower values where there are no or little such features. Afterwards, the process is repeated for each of obtained frames for a chosen number of times. In this project I chose a classic Mobile Net model which contains only two convolution layers.

The latter layer we are convolving, the more high-level features are being searched. It works similarly to human perception. To give an example, below is a very descriptive picture with features which are searched on different CNN layers. As you can see, the application of this model is face recognition. You may ask how the model knows which features to seek. If you construct the CNN from the beginning, searched features are random. Then, during training process, weights between neurons are being adjusted and slowly CNN starts to find such features which enable to meet predefined goal, i.e. to recognize successfully images from the training set.

Between described layers there are also pooling (sub-sampling) operations which reduce dimensions of resulted frames. Furthermore, after each convolution we apply a non-linear function (called ReLU) to the resulted frame to introduce non-linearity to the model.

Eventually, there are also fully connected layers at the end of the network. The last set of frames obtained from convolution operations is flattened to get a one-dimensional vector of neurons. From this point we put a standard, fully-connected neural network. At the very end, for classification problems, there is a softmax layer. It transforms results of the model to probabilities of a correct guess of each class

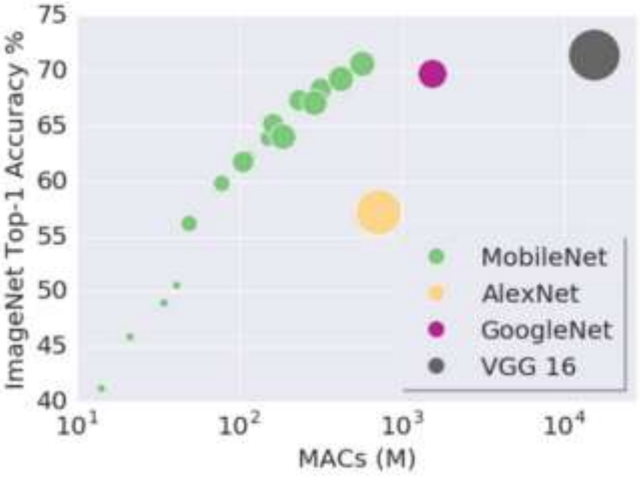
**MobileNet | CNN model**
**Architecture:**



As the name applied, the MobileNet model is designed to be used in mobile applications, and it is TensorFlow's first mobile computer vision model.
MobileNet uses depthwise separable convolutions. It significantly reduces the number of parameters when compared to the network with regular convolutions with the same depth in the nets. This results in lightweight deep neural networks.
A depthwise separable convolution is made from two operations.

1. Depthwise convolution.
2. Pointwise convolution.

MobileNet is a class of CNN that was open-sourced by Google, and therefore, this gives us an excellent starting point for training our classifiers that are insanely small and insanely fast.
The speed and power consumption of the network is proportional to the number of MACs (Multiply-Accumulates) which is a measure of the number of fused Multiplication and Addition operations.



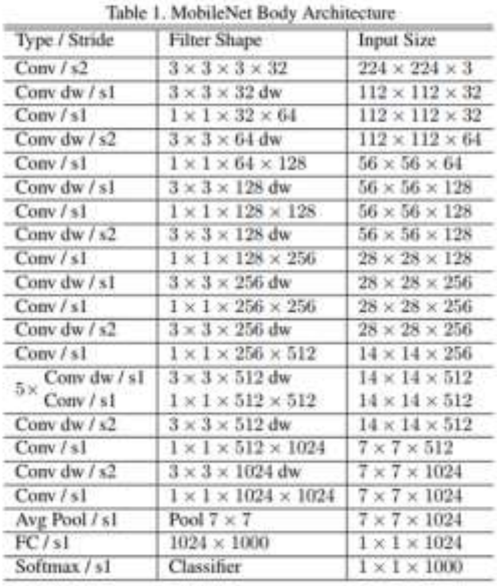
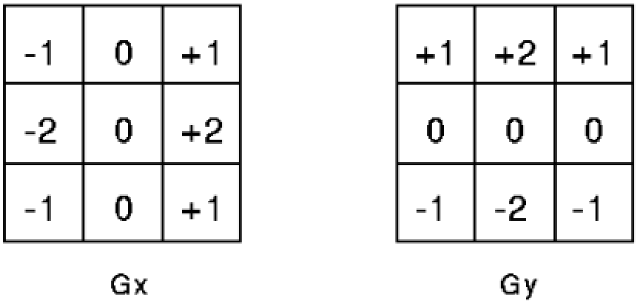


Table 1. MobileNet Body Architecture

| Type / Stride | Filter Shape | Input Size |
|---|---|---|
| Conv / s2 | 3 × 3 × 3 × 32 | 224 × 224 × 3 |
| Conv dw / s1 | 3 × 3 × 32 dw | 112 × 112 × 32 |
| Conv / s1 | 1 × 1 × 32 × 64 | 112 × 112 × 32 |
| Conv dw / s2 | 3 × 3 × 64 dw | 112 × 112 × 64 |
| Conv / s1 | 1 × 1 × 64 × 128 | 56 × 56 × 64 |
| Conv dw / s1 | 3 × 3 × 128 dw | 56 × 56 × 128 |
| Conv / s1 | 1 × 1 × 128 × 128 | 56 × 56 × 128 |
| Conv dw / s2 | 3 × 3 × 128 dw | 56 × 56 × 128 |
| Conv / s1 | 1 × 1 × 128 × 256 | 28 × 28 × 128 |
| Conv dw / s1 | 3 × 3 × 256 dw | 28 × 28 × 256 |
| Conv / s1 | 1 × 1 × 256 × 256 | 28 × 28 × 256 |
| Conv dw / s2 | 3 × 3 × 256 dw | 28 × 28 × 256 |
| Conv / s1 | 1 × 1 × 256 × 512 | 14 × 14 × 256 |
| 5× Conv dw / s1 | 3 × 3 × 512 dw | 14 × 14 × 512 |
| 5× Conv / s1 | 1 × 1 × 512 × 512 | 14 × 14 × 512 |
| Conv dw / s2 | 3 × 3 × 512 dw | 14 × 14 × 512 |
| Conv / s1 | 1 × 1 × 512 × 1024 | 7 × 7 × 512 |
| Conv dw / s2 | 3 × 3 × 1024 dw | 7 × 7 × 1024 |
| Conv / s1 | 1 × 1 × 1024 × 1024 | 7 × 7 × 1024 |
| Avg Pool / s1 | Pool 7 × 7 | 7 × 7 × 1024 |
| FC / s1 | 1024 × 1000 | 1 × 1 × 1024 |
| Softmax / s1 | Classifier | 1 × 1 × 1000 |

## Depthwise Separable Convolution

This convolution originated from the idea that a filter's depth and spatial dimension can be separated- thus, the name separable. Let us take the example of Sobel filter, used in image processing to detect edges.



| -1 | 0 | +1 |
|---|---|---|
| -2 | 0 | +2 |
| -1 | 0 | +1 |

Gx

| +1 | +2 | +1 |
|---|---|---|
| 0 | 0 | 0 |
| -1 | -2 | -1 |

Gy

Sobel Filter. Gx for the vertical edge, Gy for horizontal edge detection

You can separate the height and width dimensions of these filters. Gx filter can be viewed as a matrix product of [1 2 1] transpose with [-1 0 1].

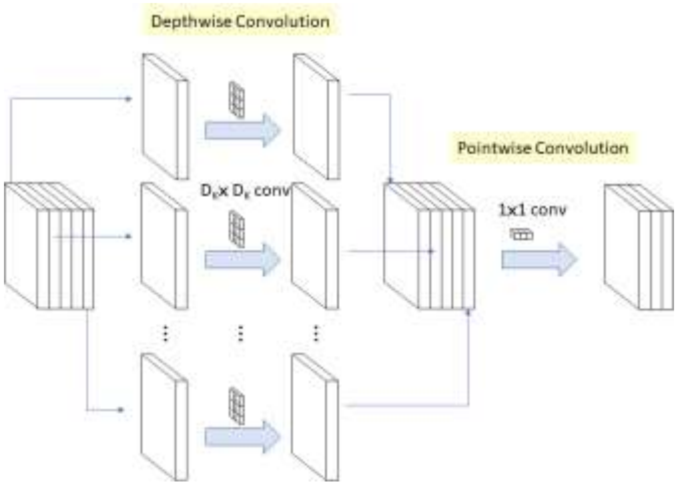
We notice that the filter had disguised itself. It shows it had nine parameters, but it has 6. This has been possible because of the separation of its height and width dimensions.

The same idea applied to separate depth dimension from horizontal (width*height) gives us depth-wise separable convolution whare we perform depth-wise convolution. After that, we use a 1*1 filter to cover the depth dimension.

One thing to notice is how much parameters are reduced by this convolution to output the same no. of channels. To produce one channel, we need 3*3*3 parameters to perform depth-wise convolution and 1*3 parameters to perform further convolution in-depth dimension.

But If we need three output channels, we only need 31*3 depth filter, giving us a total of 36 ( = 27 +9) parameters while for the same no. of output channels in regular convolution, we need 33*3*3 filters giving us a total of 81 parameters.

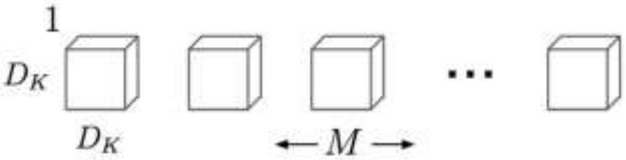
Depthwise separable convolution is a depthwise convolution followed by a pointwise convolution as follows:



Depthwise Separable Convolution

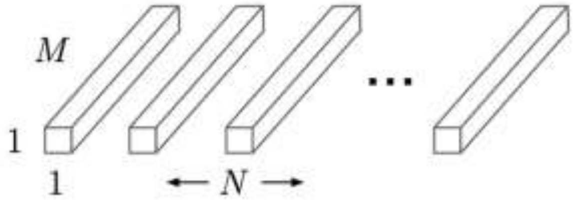
1. Depthwise convolution is the channel-wise DK×DK spatial convolution. Suppose in the figure above, and we have five channels; then, we will have 5 DK×DK spatial convolutions.
2. Pointwise convolution is the 1×1 convolution to change the dimension.
3. Depthwise convolution.

### 1. Depthwise convolution.



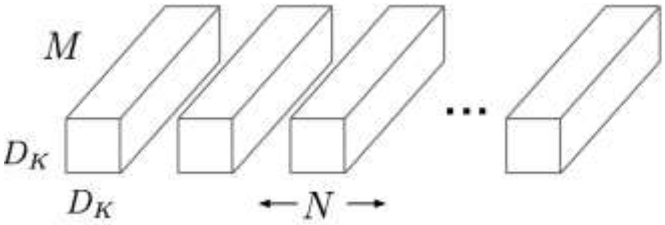
It is a map of a single convolution on each input channel separately. Therefore its number of output channels is the same as the number of the input channels. Its computational cost is **Df² * M * Dk².**

### 2. Pointwise convolution.



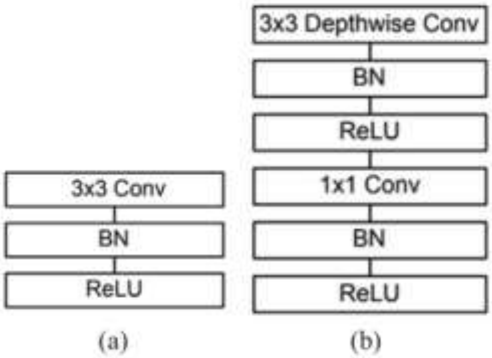
Convolution with a kernel size of 1x1 that simply combines the features created by the depthwise convolution. Its computational cost is **M * N * Df².**

Difference between Standard Convolution and Depthwise separable convolution.



**Standard Convolution.**

The main difference between MobileNet architecture and a traditional CNN instead of a single 3x3 convolution layer followed by the batch norm and ReLU. Mobile Nets split the convolution into a 3x3 depth-wise conv and a 1x1 pointwise conv, as shown in the figure.



## IV. METHODOLOGY

### DATA ACQUISITION:

In the first module of Detection of Cardiovascular Diseases in ECG, we developed the system to get the input dataset. Data collection process is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get; the better our model will perform. There are several techniques to collect the data, like web scraping, manual interventions. Our dataset is placed in the project and it's located in the model folder. The dataset is referred from the popular standard dataset repository kaggle where all the researchers refer it. The dataset consists of 1377 ECG images. The following is the URL for the dataset referred from kaggle.

We will be using Python language for this. First we will import the necessary libraries such as keras for building the main model, sklearn for splitting the training and test data, PIL for converting the images into array of numbers and other libraries such as pandas, numpy, matplotlib and tensorflow.

### Data pre-processing:

In this module we will retrieve the images from the dataset and convert them into a format that can be used for training and testing the model. This involves reading the images, resizing them, and normalizing the pixel values. We will retrieve the images and their labels. We will retrieve the images and their labels. Then resize the images to (224,224) as all images should have same size for recognition. Then convert the images into numpy array.

## FEATURE EXTRACTION AND SCALLING

Feature extraction identifies discriminative signal characteristics crucial for deep learning algorithms. This process is vital in preparing data for consumption by models, particularly in Convolutional Neural Networks (CNNs).

In CNNs, feature extraction involves the utilization of convoluted layers to scan images multiple times, capturing essential features crucial for accurate classification.

Each convolutional operation results in a set of new frames containing information about specific features present in the scanned image, forming the basis for subsequent analysis and classification.

The scanning process is characterized by parameters like stride and padding type, influencing the spatial coverage and granularity of feature extraction.

Feature extraction in CNNs is followed by operations such as max-pooling to reduce dimensionality and non-linear activation functions like ReLU to introduce non-linearity.

Feature Scaling:

Feature scaling is a preprocessing step that normalizes the range of features in a dataset, improving optimization and training efficiency in machine learning algorithms.

By scaling features to a common range, feature scaling ensures that no particular feature dominates the learning process due to differences in magnitude.

In CNNs, feature scaling aids in accelerating training by making the optimization landscape smoother and reducing the likelihood of convergence issues.

Common techniques for feature scaling include standardization (z-score normalization) and min-max scaling, ensuring consistent data representation across features.

Feature scaling enhances model performance and convergence rates, particularly in deep learning architectures like CNNs, where efficient optimization is critical for accurate classification.

Feature extraction identifies discriminative signal characteristics, crucial for deep learning algorithms. In CNNs, convoluted layers play a vital role in this process, extracting features from input data such as images or signals.

Feature scaling is the normalization of feature ranges within a dataset. It's essential for optimization, enhancing training efficiency by ensuring that features contribute uniformly to the learning process. This normalization facilitates faster convergence during training, improving the model's ability to generalize to unseen data.

The integration of MobileNet architecture in CNNs introduces depthwise and pointwise convolutions, optimizing feature extraction by reducing computational complexity. These convolutions efficiently capture spatial and channel-wise information, enhancing the model's ability to extract relevant features from input data.

Depthwise separable convolutions, a key component of MobileNet architecture, consist of two operations: depthwise convolution and pointwise convolution. This separation significantly reduces the number of parameters, enhancing computational efficiency without sacrificing performance.

Pointwise convolution, another component of MobileNet, operates on the output of depthwise convolution, efficiently combining features across channels. By decoupling spatial and channel-wise convolutions, MobileNet optimizes feature extraction, leading to more efficient and lightweight CNN models.

These details highlight the significance of feature extraction and scaling in deep learning, particularly in the context of MobileNet architecture's advancements in optimizing these processes for improved model performance and efficiency.

## SPLITTING THE DATA SET

The dataset is divided into training and testing sets, typically with an 80-20 split, where 80% of the data is used for training and 20% for testing. This division allows the model to learn patterns from the training data and evaluate its performance on unseen data.

The training data is used to train the model, while the testing data is used to assess the model's performance on data it hasn't seen before. This helps to evaluate how well the model generalizes to new, unseen examples.

Splitting the dataset ensures that the model's performance is evaluated on independent data, providing a more accurate assessment of its effectiveness in real-world scenarios.

The split between training and testing data is crucial for training machine learning models and assessing their performance. It helps prevent overfitting by ensuring that the model doesn't memorize the training data and performs well on new, unseen data.

These details underscore the importance of splitting the dataset into training and testing sets to effectively train and evaluate machine learning models.

## MODEL DEVELOPMENT MODULE

The development of a model for the diagnosis of Cardiovascular Diseases (CVDs) using deep learning techniques represents a significant endeavor in the field of medical diagnostics. With the prevalence of heart-related conditions posing a major global health challenge, there's a pressing need for accurate and efficient methods of detection to improve patient outcomes and reduce mortality rates. In this context, the utilization of electrocardiogram (ECG) signals as a diagnostic tool has garnered considerable attention due to its non-invasive nature and widespread availability in clinical settings.

The proposed model aims to leverage the power of deep learning, particularly the MobileNet Architecture, to enhance the detection of cardiovascular diseases from ECG signals. The choice of architecture is crucial as it dictates the model's capacity to extract relevant features from the input data and make accurate predictions. In this case, MobileNet Architecture is selected for its efficiency and effectiveness in processing image data, making it well-suited for analyzing ECG tracings.

The first step in model development involves the acquisition and preprocessing of the dataset. The dataset used in this project consists of ECG images obtained from cardiac patients, categorized into four classes: Normal person (NP), Abnormal Heartbeat (AH), Myocardial Infarction (MI), and History of Myocardial Infarction (H. MI). This dataset serves as the foundation for training and testing the proposed model, providing the necessary input data for learning and evaluation.

Utilizing the MobileNet Architecture, the model is designed to detect cardiovascular diseases from images of ECG tracings and predict different types of cardiac arrhythmias. The architecture comprises multiple layers of convolutional neural networks (CNNs), including 1D convolutional layers followed by batch normalization, Rectified Linear Unit (ReLU) activation, and max-pooling operations. This hierarchical structure enables the model to progressively extract features from the input data, capturing both local and global patterns relevant to cardiovascular disease diagnosis.

One of the distinguishing features of the proposed model is its avoidance of fully connected layers for classification. Instead, the model utilizes an average pooling layer followed by a softmax layer for classification. This approach eliminates the need for flattening the feature maps into a vector before feeding them into fully connected layers, reducing the computational complexity of the model and enhancing its efficiency.

Performance analysis of the model is conducted using standard evaluation metrics, including Accuracy, Precision, Recall, and F1 score. These metrics provide insights into the model's ability to correctly classify different classes of cardiovascular diseases from ECG data. By analyzing the results obtained from the confusion matrix, the model's performance can be assessed in terms of its ability to distinguish between normal and abnormal ECG patterns, as well as identify specific types of cardiac arrhythmias.

In summary, the development of a model for cardiovascular disease detection using deep learning techniques represents a promising approach to improving diagnostic accuracy and efficiency. By leveraging the MobileNet Architecture and employing rigorous performance evaluation, the proposed model aims to advance the field of cardiovascular diagnostics and contribute to better patient care.

## MODEL EVALUATION MODULE

Once the model is built, it will be applied to the validation set to evaluate its accuracy and loss. The accuracy and loss will be plotted as a function of the number of epochs to visualize the performance of the model. We will compile the model and apply it using fit function. The batch size will be 10. Then we will plot the graphs for accuracy and loss. We got average training accuracy of 97.00%. After training and evaluating the model on the validation set, the accuracy of the model will be assessed on the test set. The accuracy on the test set will be an important metric for evaluating the model's performance. We got an accuracy of 91.00% on test set.

Validation Set Evaluation: Once the model is built, it undergoes evaluation on a validation set to assess its accuracy and loss. This evaluation aids in understanding the model's performance during training and helps in tuning hyperparameters.

Training and Testing: The model is compiled and applied using the fit function, with a specified batch size (e.g., 10). Training and testing data are split, typically with 80% used for training and 20% for testing, to ensure the model's generalization ability.

Performance Visualization: Graphs for accuracy and loss are plotted as functions of the number of epochs to visualize the model's performance over time. These graphs provide insights into how the model learns and improves over successive iterations.

Accuracy Assessment: The accuracy of the model is calculated based on its performance on the test set. This metric serves as a key indicator of the model's effectiveness in correctly classifying cardiovascular diseases from ECG data.

Results: In a specific instance, an average training accuracy of 97.00% was achieved, indicating the model's ability to learn patterns from the training data. Additionally, an accuracy of 91.00% was attained on the test set, demonstrating the model's capability to generalize to unseen data.

Overall, the model evaluation module plays a crucial role in assessing the performance and effectiveness of the developed model for cardiovascular disease detection. Through rigorous evaluation and analysis of key metrics, such as accuracy and loss, the module provides valuable insights into the model's capabilities and limitations, ultimately guiding further refinement and optimization efforts.

## SAVING THE TRAINED MODEL

Once you’re confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or .pkl file using a library like pickle.

Make sure you have pickle installed in your environment.
Next, let’s import the module and dump the model into .h5 file.
Saving into .h5 or .pkl File: Once the model is trained and tested, it needs to be saved into a file format like .h5 (HDF5) or .pkl (pickle) using a library such as pickle.
Library Requirement: To perform this saving operation, the pickle library needs to be installed in the development environment.
Module Importing: The pickle module is imported into the Python environment to utilize its functionality for saving the trained model.
Dumping the Model: Using the pickle.dump() function, the trained model is dumped into a .h5 file. This file format is commonly used for saving trained neural network models in TensorFlow.
Overall, this process ensures that the trained model can be easily retrieved and used for inference or further analysis without needing to retrain the model from scratch each time.

## USER INTERFACE MODEL

Purpose: The user interface module serves as the front-end component of the cardiovascular disease detection system, providing an intuitive and interactive interface for users to interact with the trained model.
Development: The user interface is developed using software tools and frameworks tailored to create graphical user interfaces (GUIs), facilitating ease of use and seamless interaction for end-users.
Software Requirements: Several software tools and libraries are essential for developing the user interface module, including:
Python: As the primary programming language for developing the interface logic and integrating with other components.
GUI Frameworks: Frameworks such as Tkinter, PyQt, or wxPython are commonly used to create GUI elements such as buttons, text boxes, and dropdown menus.
Web Technologies: For web-based interfaces, HTML, CSS, and JavaScript are utilized along with frameworks like Flask or Django for backend integration.
Data Visualization Libraries: Libraries like Matplotlib or Plotly may be employed for displaying visualizations or graphs related to model outputs or performance metrics.
Machine Learning Integration: Integration with the trained model requires libraries such as TensorFlow or PyTorch to load the model and perform inference on user input data.
Functionality: The user interface provides functionalities such as:
Input Data: Allowing users to input ECG signals or relevant patient data for disease detection.
Prediction Output: Displaying the model's prediction results, such as the probability of different cardiovascular diseases based on the input data.
Visualization: Visualizing model performance metrics, such as accuracy, precision, recall, and F1 score, to provide insights into the model's efficacy.

## RESULTS AND FINDINGS

Model Performance Metrics: The performance of the developed model was evaluated using various metrics, including accuracy, precision, recall, and F1 score. These metrics provide quantitative measures of the model's effectiveness in detecting cardiovascular diseases from ECG signals.
Accuracy: Accuracy measures the percentage of correctly classified instances out of the total instances. It

indicates how well the model performs overall in classifying ECG signals into different disease categories.
Precision: Precision represents the ratio of correctly classified positive instances to all instances classified as positive. It measures the model's ability to avoid false positives, i.e., instances incorrectly classified as positive.
Recall: Recall, also known as sensitivity, measures the ratio of correctly classified positive instances to all actual positive instances. It quantifies the model's ability to capture all instances of a particular disease class.
F1 Score: The F1 score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance. It considers both false positives and false negatives and is particularly useful for imbalanced datasets.
Training and Testing Results: Specific results obtained during model evaluation were discussed, including an **average training accuracy of 97.00% and a test set accuracy of 91.00%.** These results highlight the model's ability to learn patterns from the training data and generalize well to unseen data.
Graphical Visualization: Graphs depicting the accuracy and loss of the model as a function of the number of epochs were plotted. These visualizations provide insights into how the model's performance evolves during the training process.
Overall, the results demonstrate the effectiveness of the developed model in accurately detecting cardiovascular diseases from ECG signals. The combination of high accuracy, precision, recall, and F1 score reflects the model's robustness and reliability in clinical applications.

## V. CONCLUSION AND FUTURE WORK

In this study, we proposed a lightweight CNN-based model to classify the four major cardiac abnormalities: abnormal heartbeat, myocardial infarction, history of myocardial infarction, and normal person classes using public ECG images dataset of cardiac patients. According to the results of the experiments, the proposed MobileNet Architecture achieves remarkable results in cardiovascular disease classification and can also be used as a feature extraction tool for the traditional machine learning classifiers. Thus, the proposed CNN model can be used as an assistance tool for clinicians in the medical field to detect cardiac diseases from ECG images and bypass the manual process that leads to inaccurate and time-consuming results.
FUTURE WORK:
In future work, optimization techniques can be used to obtain optimized values for the hyperparameters of the proposed CNN model. The proposed model can also be used for predicting other types of problems. Since, the proposed model belongs to the family of low-scale deep learning methods in terms of the number of layers, parameters, and depth. Therefore, a study on using the proposed model in the Industrial Internet of Things (IIoT) domain for classification purposes can be explored

## REFERENCES

1. "Cardiovascular diseases," World Health Organization (WHO), 11. 06. 2021. [Online]. Available: https://www.who.int/health-topics/cardiovascular-diseases.
2. "Common medical tests to diagnose heart conditions," Government of Westren Australia, Department of Health, [Online]. Available: https://www.healthywa.wa.gov.au/Articles/A_E/Common-medical-tests-to-diagnose-heart-conditions.

3. M. Swathy and K. Saruladha, "A comparative study of classification and prediction of Cardio-Vascular Diseases (CVD) using Machine Learning and Deep Learning techniques," ICT Express, 2021. https://doi.org/10.1016/j.icte.2021.08.021..
4. R. R. Lopes, H. Bleijendaal, L. A. Ramo, T. E. Verstraelen, A. S. Amin, A. A. Wilde, Y. M. Pinto, B. A. de Mol and H. A. Marquering, "Improving electrocardiogram-based detection of rare genetic heart disease using transfer learning: An application to phospholamban p.Arg14del mutation carriers," Computers in Biology and Medicine, vol. 131, no. 104262, 2021. https://doi.org/10.1016/j.compbiomed.2021.104262.
5. R. J. Martis, U. R. Acharya and H. Adeli, "Current methods in electrocardiogram characterization," Computers in Biology and Medicine, vol. 48, pp. 133-149, 2014. https://doi.org/10.1016/j.compbiomed.2014.02.012..
6. A. Rath, D. Mishra, G. Panda and S. C. Satapathy, "Heart disease detection using deep learning methods from imbalanced ECG samples," Biomedical Signal Processing and Control, vol. 68, no. 102820, 2021. https://doi.org/10.1016/j.bspc.2021.102820..
7. A. Mincholé and B. Rodriguez, "Artificial intelligence for the electrocardiogram," Nature Medicine, vol. 25, no. 1, pp. 22-23, 2019. https://doi.org/10.1038/s41591-018-0306-1.
8. A. Isin and S. Ozdalili, "Cardiac arrhythmia detection using deep learning," Procedia Computer Science, vol. 120, pp. 268-275, 2017. https://doi.org/10.1016/j.procs.2017.11.238..
9. H. Bleijendaal, L. A. Ramos, R. R. Lopes, T. E. Verstraelen, S. W. E. Baalman, M. D. Oudkerk Pool, F. V. Y. Tjong, F. M. Melgarejo-Meseguer, J. Gimeno-Blanes, J. R. Gimeno-Blanes, A. S. Amin, M. M. Winter, H. A. Marquering, W. E. M. Kok, A. H. Zwinderman, A. A. M. Wilde and Y. M. Pinto, "Computer versus Cardiologist: Is a machine learning algorithm able to outperform an expert in diagnosing phospholamban (PLN) p.Arg14del mutation on ECG?," Heart rhythm., vol. 18, no. 1, pp. 79-87, 2020. https://doi.org/10.1016/j.hrthm.2020.08.021.
10. U. R. Acharya, H. Fujita, O. S. Lih, M. Adam, J. H. Tan and C. K. Chua, "Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network," Knowledge-Based Systems, vol. 132, pp. 62-71, 2017. https://doi.org/10.1016/j.knosys.2017.06.003..
11. M. Kantardzic, Data Mining: Concepts, Models, Methods, and Algorithms, 3 ed., John Wiley & Sons, Inc., 2020.
12. S. García, J. Luengo and F. Herrera, Data Preprocessing in Data Mining, 1 ed., Springer, 2015.
13. G. Dougherty, Pattern Recognition and Classification: An Introduction, Springer, 2013.
14. A. Subasi, Practical Machine Learning for Data Analysis Using Python, Academic Press, 2020.
15. J. Soni, U. Ansari, D. Sharma and S. Soni, "Predictive data mining for medical diagnosis: An overview of heart disease prediction," International Journal of Computer Applications, vol. 17, no. 8, pp. 43-48, 2011.
16. K. Dissanayake and M. G. Md Johar, "Comparative Study on Heart Disease Prediction Using Feature Selection Techniques on Classification Algorithms," Applied Computational Intelligence and Soft Computing, vol. 2021, 2021. https://doi.org/10.1155/2021/5581806.
17. A. H. Gonsalves, F. Thabtah, R. M. A. Mohammad and G. Singh, "Prediction of coronary heart disease using machine learning: An experimental analysis," in Proceedings of the 2019 3rd International Conference on Deep Learning Technologies, 2019. https://doi.org/10.1145/3342999.3343015.
18. H. Kim, M. I. M. Ishag, M. Piao, T. Kwon and K. H. Ryu, "A data mining approach for cardiovascular

disease diagnosis using heart rate variability and images of carotid arteries," Symmetry, vol. 8, no. 6, 2016. https://doi.org/10.3390/sym8060047.

19. T. Ozcan, "A new composite approach for COVID-19 detection in X-ray images," Applied Soft Computing, vol. 111, 2021. https://doi.org/10.1016/j.asoc.2021.107669.
20. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size," arXiv preprint arXiv:1602.07360, 2016.
21. A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, pp. 1097-1105, 2012.
22. A. H. Khan, M. Hussain and M. K. Malik, "Cardiac Disorder Classification by Electrocardiogram Sensing Using Deep Neural Network," Complexity, vol. 2021, 2021. https://doi.org/10.1155/2021/5512243.
23. A. H. Khan and M. Hussain, "ECG Images dataset of Cardiac Patients," Mendeley Data, V2, 2021. https://doi.org/10.17632/gwbz3fsgp8.2.
24. C. Potes, P. Saman, A. Rahman and B. Conroy, "Ensemble of feature-based and deep learning-based classifiers for detection of abnormal heart sounds," in 2016 computing in cardiology conference (CinC), 2016.
25. A. Nannavecchia, F. Girardi, P. R. Fina, M. Scalera and G. Dimauro, "Personal Heart Health Monitoring Based on 1D Convolutional Neural Network," Journal of Imaging, vol. 7, no. 2, 2021. https://doi.org/10.3390/jimaging7020026.
26. Q. Zhang, D. Zhou and X. Zeng, "HeartID: A Multiresolution Convolutional Neural Network for ECG-Based Biometric Human Identification in Smart Health Applications," IEEE Access, vol. 5, pp. 11805-11816, 2017. https://doi.org/10.1109/ACCESS.2017.2707460.
27. U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam and R. S. Tan, "A deep convolutional neural network model to classify heartbeats," Computers in biology and medicine, vol. 89, pp. 389-396, 2017. https://doi.org/10.1016/j.compbiomed.2017.08.022.
28. R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande and P. Singh, "Prediction of heart disease using a combination of machine learning and deep learning," Computational Intelligence and Neuroscience, vol. 2021, 2021. https://doi.org/10.1155/2021/8387680.
29. P. Bizopoulos and D. Koutsouris, "Deep learning in cardiology," IEEE reviews in biomedical engineering, vol. 12, pp. 168-193, 2018. https://doi.org/10.1109/RBME.2018.2885714..
30. S. Kiranyaz, T. Ince and M. Gabbouj, "Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks," IEEE Transactions on Biomedical Engineering, vol. 63, no. 3, pp. 664-675, 2016. https://doi.org/10.1109/TBME.2015.2468589.
31. M. Zubair, J. Kim and . C. Yoon, "An Automated ECG Beat Classification System Using Convolutional Neural Networks," in 2016 6th International Conference on IT Convergence and Security (ICITCS), 2016. https://doi.org/10.1109/ICITCS.2016.7740310.
32. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018. arXiv:1801.04381.
33. T. Rahman, A. Akinbi, M. E. Chowdhury, T. A. Rashid, A. Şengür, A. Khandakar, K. R. Islam and A. M. Ismael, "COV-ECGNET: COVID-19 detection using ECG trace images with deep convolutional neural network," 2021. arXiv preprint arXiv:2106.00436.

34. G. Huang, Z. Liu, L. V. D. Maaten and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.
35. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.
36. A. Pal, R. Srivastva and Y. N. Singh, "CardioNet: An Efficient ECG Arrhythmia Classification System Using Transfer Learning," Big Data Research, vol. 26, p. 100271, 2021. https://doi.org/10.1016/j.bdr.2021.100271..
37. R. Avanzato and F. Beritelli, "Automatic ECG diagnosis using convolutional neural network," Electronics, vol. 9, no. 6, p. 951, 2020. https://doi.org/10.3390/electronics9060951.
38. U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan and M. Adam, "Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals," Information Sciences, vol. 415–416, pp. 190-198, 2017. https://doi.org/10.1016/j.ins.2017.06.027.
39. M. Naz, J. H. Shah, M. A. Khan, M. Sharif, M. Raza, and R. Damaševičius, “From ECG signals to images: a transformation based approach for deep learning,” Peerj Comput Sci, vol. 7, p. e386, 2021, doi: 10.7717/peerj-cs.386.
40. H. El-Amir and M. Hamdy, Deep Learning Pipeline: Building a Deep Learning Model with TensorFlow, Apress Media, 2020.
41. M. A. Hearst, S. T. Dumais, E. Osuna, . J. Platt and B. Scholkopf, "Support vector machines," IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, 1998. https://doi.org/10.1109/5254.708428..
42. M. Abubaker and W. M. Ashour, "Efficient data clustering algorithms: improvements over Kmeans," International Journal of Intelligent Systems and Applications, vol. 5, no. 3, pp. 37-49, 2013. https://doi.org/10.5815/ijisa.2013.03.04.
43. B. Charbuty and A. Abdulazeez, "Classification based on decision tree algorithm for machine learning," Journal of Applied Science and Technology Trends, vol. 2, no. 1, pp. 20-28, 2021.
44. L. Breiman , "Random forests," Machine learning, vol. 45, no. 1, pp. 5-32, 2001. https://doi.org/10.1023/A:1010933404324.
45. E. Miranda, E. Irwansyah,, A. Y. Amelga, M. M. Maribondang and M. Salim, "Detection of cardiovascular disease risk's level for adults using naive Bayes classifier," Healthcare informatics research, vol. 22, no. 3, pp. 196-205, 2016. https://doi.org/10.4258/hir.2016.22.3.196.
46. G. Masetti and F. D. Giandomenico, "Analyzing Forward Robustness of Feedforward Deep Neural Networks with LeakyReLU Activation Function Through Symbolic Propagation," in Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2020.
47. S. Shahinfar, P. Meek, and G. Falzon, "How many images do I need?” Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring, " Ecological Informatics, Vol. 57, 101085, 2020, https://doi.org/10.1016/j.ecoinf.2020.101085.
48. B. Zoph, E. D. Cubuk, G. Ghiasi, T. Lin, J. Shlens, and Q. V. Le. "Learning data augmentation strategies for object detection." In European conference on computer vision, pp. 566-583. Springer, Cham, 2020.