Deep Learning-Based Arrhythmia Classification using 2-D ECG Spectral Images.

A PROJECT REPORT

Submitted by,

CHINTA DEEKSHITH REDDY -20201CAI0175
JAJALA REDDY RAHUL -20201CAI0198
JALA SRINATH -20201CAI0209
CHEREDDY JAYA SREEKAR REDDY -20201CAI0210

Under the guidance of,

MR.AFROZ PASHA

in partial fulfillment for the award of the

degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

At

PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

CERTIFICATE

This is to certify that the Project report **“Deep Learning-Based Arrhythmia Classification using 2-D ECG Spectral Images”** being submitted by “CHINTA DEEKSHITH REDDY, JAJALA REDDY RAHUL, JALA SRINATH, CHEREDDY JAYA SREEKAR REDDY” bearing roll number “20201CAI0175, 20201CAI0198, 20201CAI0209, 20201CAI0210” in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering(Artificial Intelligence and Machine Learning) is a bonafide work carried out under my supervision.

Mr.Afroz Pasha
Assistant Professor
School of CSE
Presidency University

Dr. Zafar Ali Khan N
Associate Professor& HoD
School of CSE
Presidency University

Dr. C. KALAIARASAN
Associate Dean
School of CSE&IS
Presidency University

Dr. SHAKKEERA L
Associate Dean
School of CSE&IS
Presidency University

Dr. SAMEERUDDIN KHAN Dean
School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled "**Deep Learning-Based Arrhythmia Classification using 2-D ECG Spectral Images"** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Mr.Afroz Pasha, Assistant Professor, School of Computer Science and Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Name	**Roll No**	**Signature**
Chinta Deekshith Reddy	**20201CAI0175**	
Jajala Reddy Rahul	**20201CAI0198**	
Jala Srinath	**20201CAI0209**	
Chereddy Jaya Sreekar Reddy	**20201CAI0210**	

ABSTRACT

This task makes a speciality of the development of an innovative approach to Arrhythmia classification by using leveraging deep studying techniques and a couple of 2-D spectral pictures derived from electrocardiogram (ECG) facts. The primary objective is to cope with the demanding situations related to accurate arrhythmia category and prognosis, in the end contributing to improved patient outcomes and the development of cardiac health diagnostics. The significance of this project lies in its capability to revolutionize the field of cardiac fitness by way of imparting numerous key advantages, together with stronger accuracy in arrhythmia type, timely intervention in instances of abnormal coronary heart rhythms, advanced healthcare performance, and the capacity for telemedicine packages. By automating the arrhythmia class method, this mission goals to streamline the diagnostic workflow and facilitate remote tracking and prognosis of cardiac situations. In addressing studies gaps in current techniques, the undertaking seeks to triumph over demanding situations related to records heterogeneity and variety, interpretability of deep mastering fashions, robustness to noise and artifacts in ECG alerts, small sample length and imbalanced statistics, medical validation and adoption, ethical and prison considerations, as well as actual-time processing and deployment. By exploring these areas, the mission objectives to increase extra sturdy, accurate, and clinically applicable deep learning-based arrhythmia classification systems. The proposed system will combine superior deep mastering fashions skilled on various and representative datasets of 2-D spectral pictures. These fashions could be carefully demonstrated and tested to make certain their accuracy and generalizability in classifying exceptional styles of Arrhythmias. The closing aim is to broaden a robust and reliable machine that could aid healthcare providers inside the accurate and timely diagnosis of Arrhythmias, leading to improved affected person outcomes and higher control of cardiac fitness. The importance of this challenge lies in its capacity to revolutionize the sphere of cardiac fitness by using offering more desirable accuracy, well timed intervention, advanced healthcare efficiency, and capacity for telemedicine. The challenge addresses studies gaps in present methods, which include data heterogeneity, interpretability, robustness to noise, small pattern size, medical validation, moral and prison considerations, and actual-time processing and deployment.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean of School of Computer Science and Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. Kalaiarasan C and Dr. Shakkeera L,** School of Computer Science Engineering & Information Science, Presidency University and **Dr. Zafar Ali Khan**, Head of the Department, School of Computer Science & Engineering, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Mr.Afroz Pasha**,Assistant Professor, School of Computer Science and Engineering, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

Coordinators **Dr. Sanjeev P Kaulgud, Dr. Mrutyunjaya MS** and also the department Project Coordinators **Dr.Murali Parameswaran.**
We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

Chinta Deekshith Reddy(20201CAI0175)
Jajala Reddy Rahul (20201CAI0198)
Jala Srinath (20201CAI0209)
Chereddy Jaya Sreekar Reddy(20201CAI0210)

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

The Deep Learning-Based Arrhythmia Classification using 2-D ECG Spectral aims to broaden an advanced gadget for the automated type of Arrhythmias using deep mastering strategies and a pair of 2-D spectral images derived from electrocardiogram (ECG) facts. This mission is motivated by way of the want for correct and green strategies of diagnosing and classifying strange coronary heart rhythms, which might be critical for well timed and effective medical intervention.

Arrhythmia is a type of heart disease and refers to irregular changes in heart rate. There are many types of Arrhythmias, including atrial fibrillation, premature beats ventricular fibrillation, and tachycardia. Although an Arrhythmia does not have a serious impact on life, persistent Arrhythmias can be fatal. In this project, we use convolutional neural networks (CNN) to create an effective electrocardiogram (ECG) arrhythmia classification method and classify ECG into six categories; one category is normal and another five are different groups.

Treatment of cardiac Arrhythmias using deep 2-D CNN and grayscale electrocardiogram images. We are creating a web application where users can choose the images to share. Photos are provided in the training and the reference list will appear on the web page.

1.1. <u>How it works</u>:

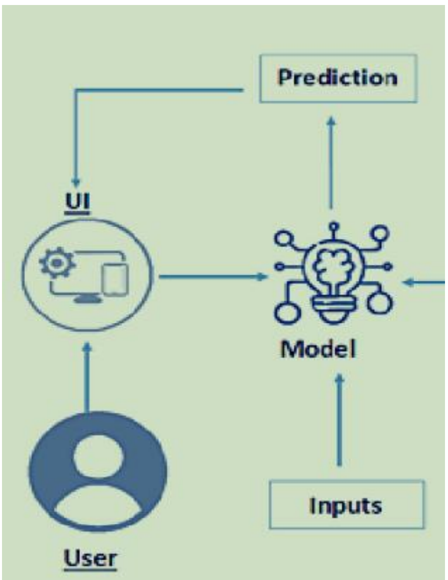


Fig-1.1:web application Working

Here we create a web application using HTML, CSS and JAVASCRIPT as front-end where user can have good user experience(UX) and flask as back-end to connect these python files and HTML,CSS and JAVASCRIPT files and deployed using Firebase. Now the a user can give a feature extracted image as input data to the designed (UI)user interface then that works on backend using given model of the processing and then we get prediction as output.

1.2.<u>The project involves the following key components:</u>

I. **Data Collection and Preprocessing**: ECG data is gathered from sufferers and preprocessed to generate 2-D spectral pix the use of sign processing strategies consisting of Fourier rework or wavelet transform. This step ensures that the ECG data is transformed into a format suitable for deep getting to know evaluation.

II. **Deep Learning Model Development**: Deep gaining knowledge of fashions, including convolutional neural networks (CNNs), are developed and skilled on the 2-D spectral snap shots to automatically examine and extract capabilities associated with exceptional sorts of Arrhythmias. The models are optimized to gain excessive accuracy in classifying Arrhythmias.

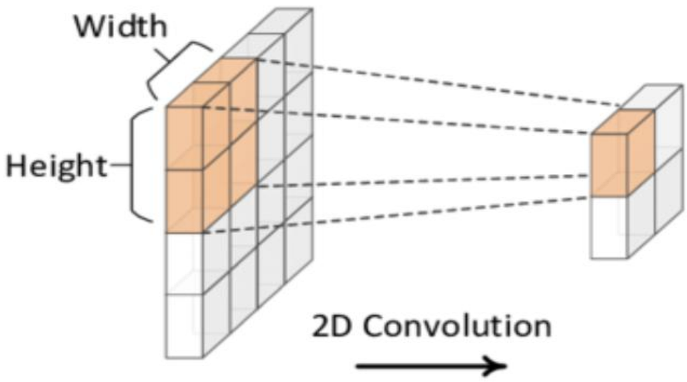


Fig 1.2.1: 2-D CNN(Convolution Neural Networks)

III. **Model Validation and Testing**: The educated models are fastidiously verified and tested the usage of separate datasets to assess their overall performance in accurately classifying Arrhythmias. This step ensures that the models generalize nicely to new, unseen records.

IV. **Integration and Deployment**: The verified deep mastering fashions are integrated right into a device that can technique new ECG spectral snap shots and classify them into unique arrhythmia classes. This system has the ability to assist healthcare experts in diagnosing and treating sufferers with cardiac situations.

V. **Enhanced Accuracy**: By harnessing the strength of deep learning, the venture goals to obtain a high stage of accuracy in classifying exceptional kinds of Arrhythmias. This can result in extra specific diagnoses and treatment plans for patients with cardiac conditions.

VI. **Timely Intervention**: The automated class machine has the potential to expedite the analysis method, making an allowance for timely medical intervention in instances of ordinary coronary heart rhythms. This may be crucial in emergency conditions and in dealing with persistent cardiac conditions.

VII. **Potential for Telemedicine**: The development of an automated arrhythmia classification system could facilitate remote monitoring and diagnosis of cardiac conditions, enabling telemedicine applications for patients in remote or underserved areas.

VIII. **Improved Healthcare Efficiency**: By automating the arrhythmia classification process, healthcare providers may be able to streamline the diagnostic workflow, leading to more efficient use of resources and improved patient care.The remaining intention of this mission is to increase a robust and reliable device that may aid healthcare providers in the correct and well timed analysis of Arrhythmias, main to improved affected person outcomes and better control of cardiac fitness.

Ambulatory electrocardiogram (ECG) about the size of a postcard or a digital camera that the patient will use up to 2 days in a weeks. The test measures the movement of electrical signals or waves through the heart. These signals tell the heart to contract (squeeze) and pump blood. The patient will have electrodes taped to their skin. His painless, although some people experience mild skin irritation from the tape used to attach electrodes to the chest.

They can do everything except shower or bath while wearing electrodes. After the trial period, the patient leaves back to your doctor. They will download information.

Role of Deep Learning:

Deep learning is a branch of machine learning that is entirely based on artificial intelligence neural network because neural network will mimic the human brain so deep learning it is also a kind of mimic of the human brain. In deep learning, we don't need it explicitly program everything. The concept of deep learning is not new. That was about a for a few years now. It's all the rage these days because we didn't have that much before computing power and large amounts of data. As in the last 20 years, computing power is growing exponentially, deep learning and machine learning have arrived on the scene.

Deep learning is a special type of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept Defined in relation to simpler concepts, and enumerated in terms of more abstract representations less abstract.

Convolutional Neural Networks (CNN):

In the field of deep learning, Convolutional Neural Networks (CNN) are in a class Deep Neural Networks, which were mostly deployed in the field of analysis/imagery recognition.Convolutional Neural uses a very special type of method that is happening known as convolution.

Convolutional Neural Networks (CNN) consists of different layers of artificial neurons. Artificial neurons are similar to the neurons used by the human brain There are mathematical functions, passing various sensory input signals and other responses which is used to calculate the sum of various inputs and give an output in The form of the activation value.The behavior of each CNN neuron is defined by its weight value. When Fed with values (of pixels), the artificial neurons of the CNN identify Various visual features and specifications.

When we give an input image to a CNN, each of its inner layers generates a different one Activation maps. Activation maps show the relevant features of a given input Image Each

CNN neuron typically takes input in the form of a group/patch The pixels multiply their values (colors) by their weight values, add them, and Input them through the respective activation function.

The first (or perhaps lower) level of a CNN typically identifies various features Edges of input image such as horizontal, vertical and diagonal. The output of the first layer is given as the input of the next layer, which will in turn Extract other complex features of the input image such as corners and combinations the edge. The deeper one goes into a convolutional neural network, the more layers are introduced Detecting various high-level features like objects, faces etc…

CHAPTER-2

LITERATURE SURVEY

Table 2.1: Literature Review

Reference	Year	Focus	Key Features
uardo José da S. Luz a, William Robson Schwartz b,Guillermo Cámara-Chávez a, David Menotti a,c,	2017	ECG-based heartbeat classification for arrhythmia detection.	The paper has many strengths, including the use of specialized techniques such as the wavelet transform, statistical properties, and vector cardioid graph (VCG) analysis. Additionally, the application of various machine learning algorithms such as support vector machine (SVM), neural network (ANN), and heart rate decision trees to detect arrhythmia is also discussed.
Johnson, A. et al. HybDeepNet:	2023	HybDeepNet: ECG Signal Based Cardiac Arrhythmia Diagnosis Using a Hybrid Deep Learning Model	Improved accuracy: The hybrid model shows the best performance in terms of sensitivity and specificity compared to existing standards. Noise Resistance: The 2D CNN component of the model helps improve accuracy. Effective removal methods: This model combines the advantages of CNN and LSTM models to achieve good results and improve performance.
Garcia, M. et al. Res-net period	2017-2023	Two-dimensional ECG-based cardiac	This paper provides a comprehensive review of state-

		arrhythmia classifcation using DSE updated	of-the-art deep learning methodologies for ECG Arrhythmia detection, highlighting promising approaches and recurring limitations. It offers insights into high-performing techniques and potential areas for future research and development
ResNetJiahao Li1, Shao-peng Pang1*, Fangzhou Xu2,4, Peng Ji2,4, Shuwang Zhou3,4 & Minglei Shu	2022	2-D ECG-based cardiac arrhythmia classifcation using DSE	Integrated structure that will increase classification and crime efficiency. DSE-ResNet model performed well on the CPSC2018 latent test set and achieved high F1-scores for various types of Arrhythmias..
Amin Ullah 1,2 , Syed Muhammad Anwar 1,2 , Muhammad Bilal 3 and Raja Majid Mehmood 4	2020	Remote sensing for Cardiovascular Diagnostics	The proposed model has high accuracy,sensitivity and specificity, making it reliable in diagnosing heart disease. Additionally, this study addresses the limitations of traditional methods by using two-dimensional ECG signal.
Qiao Xiao 1ORCID, Khuan Lee 2,Siti Aisah Mokhtar 1, Iskasymar Ismail	2023	Deep Learning-Based ECG Arrhythmia Classification	The strength of this article appears to be related to the in-depth analysis of deep learning models for ECG arrhythmia classification, analysis of studies, competition and opportunities for future work in

			this field, and multidisciplinary analysis of ECG.

Based on the information summarized in the research papers, it is clear that there is a strong focus on ECG arrhythmia classification using deep learning models. The papers discuss the limitations and advantages of their respective studies, highlighting the need for significant computational power, training time, and the use of only ECG signals. However, they also emphasize achievements in achieving state-of-the-art performance, high accuracy, sensitivity and specificity in heart disease diagnosis.

A literature survey on ECG arrhythmia classification using deep learning models reveals a common theme of addressing the limitations of traditional methods by leveraging two-dimensional ECG signal representation and data augmentation to improve generalization capabilities. Additionally, the papers provide insights into the challenges of classifying different classes of ECG signals and the need for further investigation to understand performance differences between patients.

Overall, the literature survey reflects the growing interest and progress in deep learning-based approaches for ECG arrhythmia classification. The studies contribute to the multi-disciplinary analysis of ECG databases, preprocessing techniques, deep learning methods, evaluation paradigms, and performance evaluation, paving the way for future work in this area.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

1. **remote sensing paper by Amin Ullah 1,2 , Syed Muhammad Anwar 1,2 , Muhammad Bilal 3 and Raja Majid Mehmood 4,* 2020**

Limitations:

Limitations of this paper include the need for significant computational power and training time to implement the CNN classifier. Additionally, only ECG signals were used alone in this study, and performance comparisons with existing methods may be limited due to changes in training and experience test data.

Advantages:

The strengths of this paper include achieving state-of-the-art performance in ECG arrhythmia classification using a CNN-based approach with 2-D spectrograms as input. The proposed model has high accuracy, sensitivity and specificity, making it reliable in diagnosing heart disease. Additionally, this study addresses the limitations of traditional methods by using two-dimensional ECG signal representation and data augmentation to improve generalization capabilities.

2. **paper 2 by Deep Learning-Based ECG Arrhythmia Classification: A Systematic Review by Qiao Xiao 1ORCID,Khuan Lee 2,Siti Aisah Mokhtar 1,Iskasymar Ismail 2023**

Limitations:

Limitations of the results paper include the mean measurement of selected studies, the difficulty of classifying different classes of ECG signals, and differences in performance between patients at the center, which will require further investigation.

Advantages:

The strength of this article appears to be related to the in-depth analysis of deep learning models for ECG arrhythmia classification, analysis of studies, competition and opportunities for future work, and multidisciplinary analysis of ECG. Databases, pre-processing techniques, deep learning methods, evaluation paradigms and performance evaluations.

3.Two-dimensional ECG-based cardiac arrhythmia classifcation using DSE-ResNetJiahao Li1, Shao-peng Pang1*, Fangzhou Xu2,4, Peng Ji2,4, Shuwang Zhou3,4 & Minglei Shu year- 2022

Limitations:

This article provides useful information on automatic classification of cardiac Arrhythmias using 2-D ECG data. However, some limitations include the increased computational complexity due to the mixed model and the need for additional validation of different data sets to evaluate generalization.

Advantages:

The paper has several advantages, including:

Using deep neural networks (DNN) to obtain classification of cardiac Arrhythmias using the ability to eliminate sources and additive learning. Integrated structure that will increase classification and crime efficiency. DSE-ResNet model performed well on the CPSC2018 latent test set and achieved high F1 scores for various types of Arrhythmias. The model is capable of recognizing patterns and learning useful features from raw data without the need for formal policies and procedures, making it suitable for the interpretation of ECG data.

4.Two-dimensional ECG-based cardiac arrhythmia classifcation using DSE-ResNet period 2017-2023

Limitations:

Limitations of this article include the focus on high-quality methods and the exclusion of studies with accuracy below 96%. Additionally, systematic reviews were not recorded and risk of bias assessment was not performed in the included studies.

Advantages:

This paper provides a comprehensive review of state-of-the-art deep learning methodologies for ECG arrhythmia detection, highlighting promising approaches and recurring limitations. It offers insights into high-performing techniques and potential areas for future research and development.

5. ECG-based heartbeat classification for arrhythmiadetection: A survey Eduardo José da S. Luz a, William Robson Schwartz b, Guillermo Cámara-Chávez ,David Menotti a,c,∗ year:2017

Limitations:

Limitations of the paper include inconsistency of classes in the data used, lack of standardized testing methods, and difficulties in fair comparison of methods due to differences in materials, paper sizes, and measurement methods. The article also highlights the challenges of creating new databases and the need for more diverse and larger datasets for fair evaluation.

Advantages:

The paper has many strengths, including the use of specialized techniques such as the wavelet transform, statistical properties, and Vector Cardioid Graph (VCG) analysis. Additionally, the application of various machine learning algorithms such as Support Vector Machine (SVM), Artificial Neural Network (ANN), and heart rate decision trees to detect arrhythmia is also discussed. Additionally, this article highlights the importance of data diversity and size for fair evaluation and proposes the use of new methods of capturing ECG signals to generate data.

6.HybDeepNet: ECG Signal Based Cardiac Arrhythmia Diagnosis Using a Hybrid Deep Learning Model year:2023

Limitations:

This article provides detailed information on the application of HybDeepNet technology for the diagnosis of cardiac Arrhythmias using ECG signals. However, there is no general discussion about the complexity of the scheme and its limitations, such as longer training and optimization.Further research could investigate these gaps to improve understanding of the applications and limitations of cognitive skills in cardiology counseling.

Advantages:

The proposed HybDeepNet technology has many advantages, including:

Improved accuracy: The hybrid model shows the best performance in terms of sensitivity and specificity compared to existing standards.

Noise Resistance: The 2D CNN component of the model helps improve accuracy by showing that it is more powerful at dealing with noise in the input signal.

Effective removal methods: This model combines the advantages of CNN and LSTM models to achieve good results and improve performance.

These results demonstrate the potential of HybDeepNet technology to detect Arrhythmias using ECG signals.

Research gaps:

A. **Data availability and quality:** Research can identify gaps in the availability and quality of ECG databases for training and testing deep learning models. Future opportunities may include efforts to address data scarcity and improve the quality and diversity of available ECG datasets.

B. **Preprocessing techniques:** There may be gaps in research and comparison of preprocessing techniques for ECG data. Future opportunities could include exploring new preprocessing methods that improve feature extraction and model performance.

C. **DL Methodology:** Research can identify gaps in the application of specific deep learning methods for ECG analysis. Future opportunities may include exploring new architectures, transfer learning approaches, or multimodel learning techniques for better ECG classification.

D. **Evaluation paradigms and performance metrics:** Gaps in the evaluation paradigms and performance metrics used to evaluate ECG classification models can be identified. Future opportunities could include standardization of assessment methods and metrics, as well as the development of new approaches to address specific challenges in ECG analysis.

E. **Code availability and reproducibility:** Research may reveal gaps in code availability and reproducibility of results in the field of ECG analysis. Future opportunities may include support for open access code repositories, reproducible research practices, and standardized benchmark datasets.

CHAPTER-4

PROPOSED METHODOLOGY

4.1. Problem Statement:

Arrhythmia is a common and potentially life-threatening condition caused by an irregular heartbeat. Accurate and timely diagnosis of Arrhythmias is crucial for effective treatment and patient care. Traditional ECG analysis has limitations in accurately classifying Arrhythmias, so more advanced classification methods are needed. This project is designed to develop a deep learning arrhythmia classification system using 2-D electrocardiogram spectrum images. The system uses a neural network model to identify and classify different patterns such as sinus disease, atrial fibrillation and ventricular tachycardia. The goal is to improve the accuracy and efficiency of arrhythmia classification, thereby improving patient outcomes and clinical decision-making.

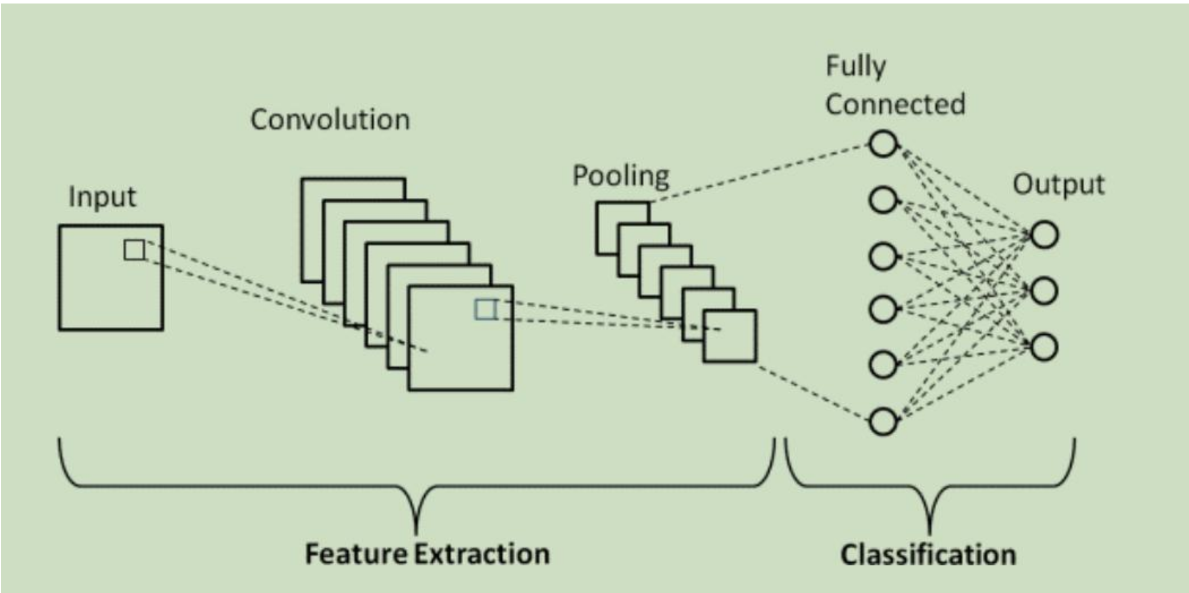


Fig 4.1:Methodology

4.2. Methodology

1. Data Collection and Curation:

- Obtain ECG statistics from numerous affected person populations, along with people with different cardiac conditions and demographics.
- Curate a complete dataset that represents a huge spectrum of Arrhythmias and ECG signal characteristics.

2. Preprocessing and Spectral Image Generation:

- Preprocess the ECG data to take away noise and artifacts, making sure the best of the input signals.
- Transform the preprocessed ECG facts into 2-D spectral images using signal processing strategies which include Fourier rework or wavelet remodel.

3. Model Development:

- Design and develop deep learning models, including convolutional neural networks (CNN's), tailored for processing 2-D spectral pix.
- Train the deep mastering fashions on the curated dataset, leveraging techniques together with switch getting to know and records augmentation to improve version generalization.

4. Validation and Testing:

- Validate the skilled fashions using rigorous pass-validation strategies to assess their accuracy and robustness throughout distinctive arrhythmia types.
- Test the models on separate datasets to evaluate their performance in correctly classifying Arrhythmias, including uncommon and imbalanced lessons.

5. Interpretability and Explainability:

- Investigate techniques for decoding and explaining the selections made by using the deep mastering models to decorate their medical application and trustworthiness.

6. Feature Extraction:

- Implement feature extraction techniques like wavelet transforms or Fourier analysis to capture relevant spectral patterns for arrhythmia detection.

7. Training Procedure:

- Train the model using appropriate loss functions and optimization algorithms, incorporating early stopping and model checkpointing.

8. Hyperparameter Tuning:

- Conduct systematic tuning of hyperparameters, optimizing learning rates, batch sizes, and network depths to enhance model performance.

9. Evaluation Metrics:

- Assess model performance using accuracy, precision, recall, F1-score, and AUC-ROC to ensure comprehensive evaluation.

10. Comparison with Baselines:

- Compare the model against baseline methods and traditional ECG analysis techniques to quantify classification accuracy improvements.

11. Clinical Validation and Integration:

- Collaborate with healthcare experts to clinically validate the overall performance of the deep studying-based totally arrhythmia type gadget.
- Integrate the established models into a consumer-friendly system which could procedure new ECG spectral photos and provide computerized arrhythmia classification.

12. Ethical and Legal Considerations:

- Address ethical and prison issues associated with affected person privateness, consent, and liability in the deployment of computerized diagnostic equipment.

13. Real-Time Processing and Deployment:

- Develop efficient algorithms and software program architecture to allow real-time processing of 2-D ECG spectral pics for well timed analysis and intervention.
- Deploy the device in medical settings, ensuring seamless integration with current healthcare infrastructure.

4.3. Convolutional Neural Network (CNN) Architecture:

Introduction:

The Deep Learning-Based Arrhythmia Classification using 2-D ECG Spectral aims to broaden an advanced gadget for the automated type of Arrhythmias using deep mastering strategies and a pair of 2-D spectral images derived from electrocardiogram (ECG) facts. This mission is motivated by way of the want for correct and green strategies of diagnosing and classifying strange coronary heart rhythms, which might be critical for well timed and effective medical intervention.

Convolutional Layers:

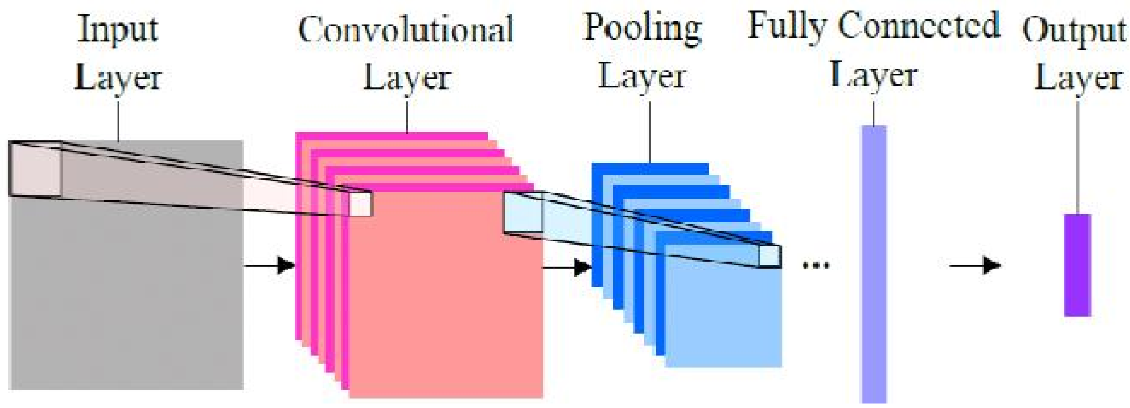


Fig 4.3.1: CNN Layers

The project involves the following key components:

1.Data Collection and Preprocessing:

ECG data is gathered from sufferers and preprocessed to generate 2-D spectral pix the use of sign processing strategies consisting of Fourier rework or wavelet transform. This step ensures that the ECG data is transformed into a format suitable for deep getting to know evaluation.

2. Deep Learning Model Development:

Deep gaining knowledge of fashions, including convolutional neural networks (CNN's), are developed and skilled on the 2-D spectral snap shots to automatically examine and extract capabilities associated with exceptional sorts of Arrhythmias. The models are optimized to gain excessive accuracy in classifying Arrhythmias.

3. Model Validation and Testing:

The educated models are fastidiously verified and tested the usage of separate datasets to assess their overall performance in accurately classifying Arrhythmias. This step ensures that the models generalize nicely to new, unseen records.

4. Integration and Deployment:

The verified deep mastering fashions are integrated right into a device that can technique new ECG spectral snap shots and classify them into unique arrhythmia classes. This system has the ability to assist healthcare experts in diagnosing and treating sufferers with cardiac situations.

5. Enhanced Accuracy:

By harnessing the strength of deep learning, the venture goals to obtain a high stage of accuracy in classifying exceptional kinds of Arrhythmias. This can result in extra specific diagnoses and treatment plans for patients with cardiac conditions.

6. Timely Intervention:

The automated class machine has the potential to expedite the analysis method, making an allowance for timely medical intervention in instances of ordinary coronary heart rhythms. This may be crucial in emergency conditions and in dealing with persistent cardiac conditions.

7. Potential for Telemedicine:

The development of an automated arrhythmia classification system could facilitate remote monitoring and diagnosis of cardiac conditions, enabling telemedicine applications for patients in remote or underserved areas.

8. Improved Healthcare Efficiency:

By automating the arrhythmia classification process, healthcare providers may be able to streamline the diagnostic workflow, leading to more efficient use of resources and improved patient care.The remaining intention of this mission is to increase a robust and reliable device that may aid healthcare providers in the correct and well timed analysis of Arrhythmias, main to improved affected person outcomes and better control of cardiac fitness.

9. Filter Sizes:

The filter length determines the spatial quantity of the functions that the filter out can locate inside the enter photograph. In the context of 2-D ECG spectral pics, the clear out length should be selected to size applicable patterns and structures in the spectral area. For example, smaller filter sizes (e.G., 3x3 or 5x5) can seize nice-grained information, while large filter sizes (e.g., 7x7 or 9x9) can capture extra global styles.

10. Number of Filters:

The wide variety of filters in a convolutional layer determines the intensity of function extraction and the range of functions that can be learned with the aid of the version. In the context of arrhythmia type, a larger range of filters can help capture a huge variety of spectral capabilities associated with one-of-a-kind arrhythmia sorts.

11. Activation Functions (e.g., ReLU):

ReLU (Rectified Linear Unit) is a generally used activation feature in CNN's because of its simplicity and effectiveness. It introduces non-linearity into the version and helps the community learn complex styles within the records. The use of ReLU as the activation feature in convolutional layers allows the version to seize non-linear relationships in the spectral snap shots, enhancing its ability to extract discriminative functions.

$$F(x) = \max(0, x)$$

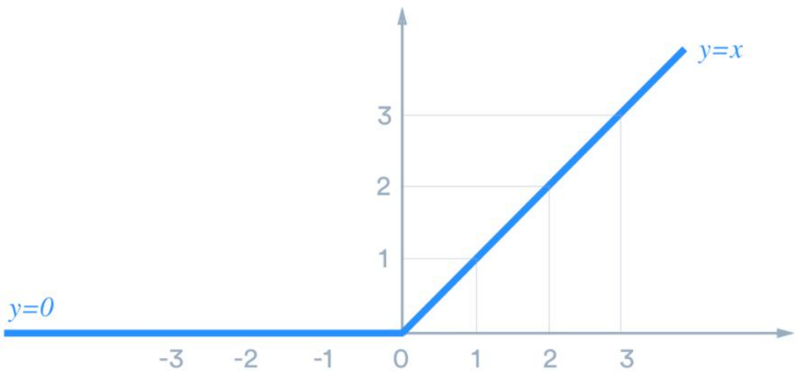


Fig 4.3.2: Activation function

Pooling Layers:

Max Pooling and Average Pooling:

In the CNN architecture for arrhythmia classification, max pooling and average pooling are usually used for spatial down sampling. Max pooling selects the maximum cost from a neighborhood location of the characteristic map, even as average pooling computes the common cost inside the identical region.

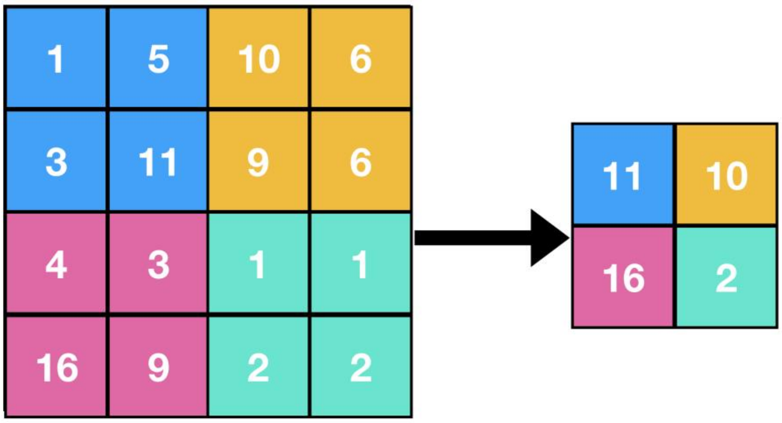


Fig 4.3.3:max pooling

Information Purpose of Pooling:

The foremost reason of pooling layers is to introduce translational in variance and decrease the spatial decision of the characteristic maps. By down sampling the characteristic maps, pooling layers assist make the learned functions greater strong to small spatial variations and decrease the computational burden of subsequent layers. Control over Over-fitting: Pooling layers can also assist manipulate over fitting by lowering the spatial dimensions of the feature maps, which can save you the version from studying noise or beside the point details within the enter information.

Downsampled Feature Maps:

After applying pooling layers, the spatial dimensions of the function maps are reduced, at the same time as crucial functions are retained. attention at the most relevant records and discard redundant or much less informative details. Justifiying the CNN depth.

Complexity and Feature Representation:

The depth of a CNN architecture, as exemplified via well-known architectures including VGG and ResNet, at once affects the version's capability to examine complex hierarchical functions from input data. A deeper architecture allows the community to capture complex patterns and representations in the spectral photographs.

Hierarchical Feature Learning:

Deeper architectures permit the CNN to learn hierarchical representations of capabilities, wherein lower layers seize simple patterns (e.g., edges, textures) and higher layers capture more abstract and complex functions applicable to arrhythmia category.

Transfer Learning Considerations:

Well-known architectures like VGG and ResNet have verified achievement in photo classification obligations and may be leveraged for switch mastering. The depth of these architectures permits for effective transfer of found out functions to the venture of arrhythmia classification.

Suitability for Spectral Images:

The chosen depth of the CNN architecture should be suitable for processing 2-D ECG spectral pics, which may also comprise intricate patterns and systems that require a deep structure to effectively capture.

Avoidance of Vanishing Gradient:

Deep architectures, while designed with suitable pass connections (as in ResNet) or cautious weight initialization, can mitigate issues including vanishing gradients, enabling powerful education and characteristic gaining knowledge of in deeper layers. The selected intensity ought to strike a stability among model complexity and computational performance. It need to be deep enough to seize relevant functions however no longer overly complex to the point of diminishing returns or computational impracticality.

Output Layer:

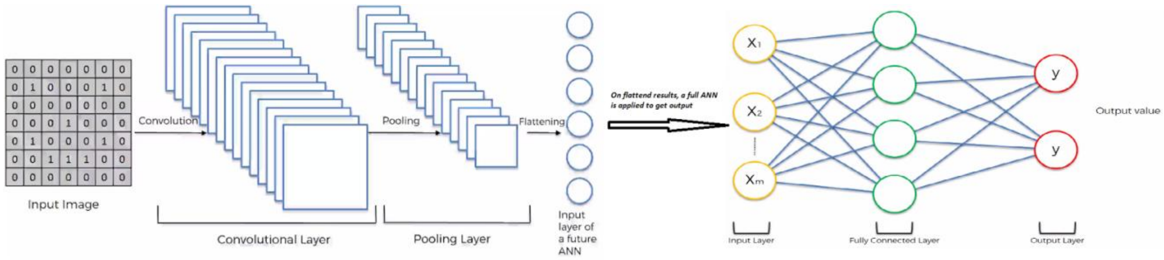


Fig 4.3.4:output layers

Configuration of the Output Layer:

The output layer of the CNN architecture for arrhythmia class is designed to produce predictions for exceptional arrhythmia training. The wide variety of neurons inside the output layer corresponds to the awesome arrhythmia classes that the model ambitions to categorize. Each neuron inside the output layer represents a specific arrhythmia elegance, and the version's prediction is based at the neuron with the very best activation.

Number of Neurons:

The range of neurons within the output layer is decided by means of the entire variety of arrhythmia instructions that the model is educated to categorise. For example, if there are N wonderful arrhythmia instructions, the output layer could have N neurons, every representing one class.

Use of Softmax Activation:

The softmax activation characteristic is usually used inside the output layer for multi-elegance type duties. It transforms the raw output rankings of each neuron into elegance probabilities, ensuring that the sum of all possibilities equals 1. This permits the model to make confident and normalized predictions throughout all lessons.

Interpretation of Softmax Output:

After making use of the softmax activation, the output of each neuron represents the chance that an input ECG spectral photograph belongs to a specific arrhythmia magnificence. The class with the very best possibility is anticipated as the maximum likely arrhythmia kind for the input picture.

4.4 Process flow:

● Users interact with the user interface to upload images

● The uploaded images are analyzed by the integrated model

● Once the model validates the uploaded images, the estimated results will appear in the user interface.Achieving this goal We must complete all the tasks and activities below.

● Data collection.

○ Gather or create information

● Prior knowledge.

○ Import ImageDataGenerator library

○ Set ImageDataGenerator class

○ Use ImageDataGenerator function for training data and test data.

● sample design

○ Import design library

○Initialize model<br< b="" style="margin: 0px; padding: 0px;"> >

○ Add input method

○ Add hidden method

○ Add output method

○ Set learning method

○ Train and test models

○ Prototype

○ Save models

● Application design

○ Creating HTML files

○ Creating Python code

● Deployment of the application

CHAPTER-5

OBJECTIVES

The objectives of the task "Deep Learning-Based Arrhythmia Classification the use of 2-D ECG Spectral Image" are designed to deal with key demanding situations in arrhythmia type and to boost the country of computerized cardiac fitness diagnostics. The goals em-body technical, scientific, and societal elements, and are geared toward achieving big advancements in the field.

1. **Develop Advanced Deep Learning Models:**
   Design and develop superior deep learning fashions, together with convolutional neural networks (CNN's), specifically tailor-made for processing 2-D spectral pix derived from ECG information.

2. **Enhance Accuracy and Generalizability:**
   Train the deep studying models on diverse and consultant datasets to gain high accuracy and generalizability in classifying one of a kind types of Arrhythmias, which include rare and imbalanced instructions.

3. **Improve Interpretability and Explainability:**
   Investigate methods for decoding and explaining the selections made through the deep getting to know models, improving their interpretability and clinical utility.

4. **Address Data Heterogeneity and Diversity:**
   Collect and curate a comprehensive dataset that represents a huge spectrum of Arrhythmias and ECG sign characteristics, making sure diversity in affected person demographics and cardiac situations.

5. **Robustness to Noise and Artifacts:**
   Develop strong preprocessing strategies and version architectures which can successfully handle noisy ECG facts at the same time as preserving excessive category accuracy.

6. Clinical Validation and Adoption:

Collaborate with healthcare experts to clinically validate the performance of the deep mastering-based totally arrhythmia category device and understand the elements that have an impact on its adoption in real-world medical practice.

7. Ethical and Legal Considerations:

Address moral implications related to affected person privacy, consent, and legal responsibility in the deployment of automatic diagnostic gear, making sure accountable use of patient statistics.

8. Enable Real-Time Processing and Deployment:

Develop green algorithms and software program structure to permit actual-time processing of 2-D ECG spectral photographs for timely prognosis and intervention, main to improved affected person effects.

9. Facilitate Telemedicine Applications:

Explore the potential for telemedicine programs by using growing a device which could facilitate far flung monitoring and diagnosis of cardiac situations, in particular in far off or underserved regions.

10. Contribute to Advancements in Cardiac Health Diagnostics:

Contribute to the improvement of clever diagnostic gear that guide healthcare specialists in providing extremely good care to patients with cardiac conditions, in the long run main to advanced control of cardiac health.

By the end of this project :

- knew fundamental concepts and techniques of the Artificial Neural Network and Convolution Neural Networks.
- Gained a broad understanding of image data.
- Worked with Sequential type of modeling.
- Worked with Keras capabilities.
- Worked with image processing techniques.
- knew how to build a web application using the Flask framework.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

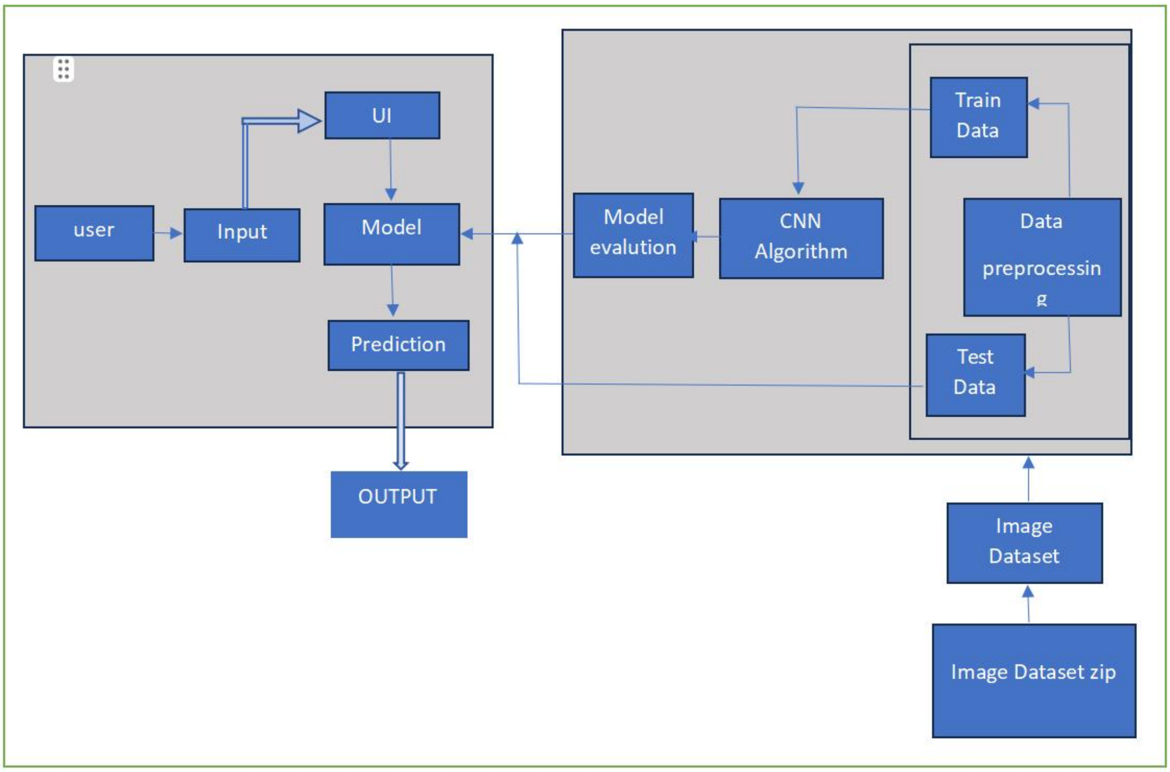


Fig 6.1: System Design & Implementation

1.UI (User Interface):

This is where customers connect to the machine, possibly to enter ECG data or capture type results.

2. Model:

This imperative block represents a deep learning model itself, possibly a convolutional neural network (CNN) skilled in classifying Arrhythmias.

3. Model Evaluation:

This block involves evaluating the version's performance using a number of metrics.

Train Data: This block contains labeled ECG data that is used to train the model.

4. Preprocessing:

This step prepares the ECG data for version input, undoubtedly related to noise removal, segmentation and normalization.

5.CNN Algorithm:

This CNN specifies a set of architecture and education rules.

6. Test Data:

This block contains invisible ECG records that are used to assess the normalization of the version.

7.Image dataset zip:

This potential stores a collection of two-day ECG spectral images.

8.Image dataset:

This block represents the extracted spectral peaks fitted to the version enter.

9. Assumptions:

This block outputs the expected arrhythmia type of the model.

System Design:

The diagram presents a system for ECG (electrocardiogram) classification using a convolutional neural network (CNN). Design consists of elements such as "user input", "UI", "model", "prediction", "output", "model evaluation", "CNN algorithm", "train data", "data preprocessing", "test data" Included ", "image dataset" and "image dataset zip"The system is designed to take user input, process it through a CNN model, make predictions and provide output to the user.

A CNN algorithm is used to process the image data, and the system involves training and testing the model using the image datasets.

Implementation:

The implementation consists of building a Flask web application to serve the ECG classification model.The application allows users to upload ECG images for prediction and provides the predicted class as output.The CNN model is implemented using TensorFlow and Keras, with data preprocessing, model generation, integration, training and prediction steps.Training a CNN model involves defining layers, setting up a data generator for training and testing datasets, and integrating the model with an appropriate optimizer and loss function.Overall, the system design and implementation involves building a machine learning system for ECG classification, including a user interface for input, a CNN model for processing, and an output mechanism for predictions. The implementation consists of creating a web application to interact with the trained model and providing functionality for users to upload ECG images for classification.

Contents:

● We are building a Flask Application that needs HTML pages stored in the templates folder and a python script app.py for serverside scripting and ECG_App for process model generation.

● we need the model which is saved and the saved model in this content is ECG.h5.

● The static folder will contain js and CSS files.

● Whenever we upload an image to predict, that images are saved in the uploads folder.

●Here we create folder called models where our model ECG.h5 is saved.

project implementation:

1. Dataset Collection

The dataset contains six classes:

1. Left Bundle Branch Block
2. Normal
3. Premature Atrial Contraction
4. Premature Ventricular Contractions
5. Right Bundle Branch Block
6. Ventricular Fibrillation

2. Image Processing:

The dataset of our project contains images of different classes (6 classes) which are required An "image processing" step before feeding the input to the ANN. This includes image processing

● Enhance image feature, image data generator library.

● Load the dataset.

● Implement augmented feature for train set and test set.

3. Model Building:

As the Image Processing step is done, now we start to build the CNN model for prediction.

This step includes the following steps:

● Import libs

● Initialize the model

● Add CNN layers

● Configure your learning

● Fit the data

● Save the model

4. Application Building:

In this project, we not only build a CNN model, but we also build an Application, a proper platform to use this trained model, using a micro web framework called "FLASK".

For this we create HTML pages (with associated CSS and JS pages) which shares some information about Arrhythmia and classifications, and in the 'predict' page, we predict the type of Arrhythmia given as Image Input.

These HTML pages are used to make a proper interface (a website) to showcase our project output. Flask folder includes

i. templates (to store the .html files needed for our website)

ii. static (to store .css .js and some images included in .html files)

iii. uploads (to store the images we uploaded while using the app)

iv. app.py (flask application development python file)

v. models (CNN trained model that is given to flask application as input which as file ECG.h5)

5. Train the model on Google Colob

• unzip the zip folder in google colob from drive.

6. Image preprocessing

7. Train class Indices

8. Model Building

9. Saving the model

CHAPTER-7
TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

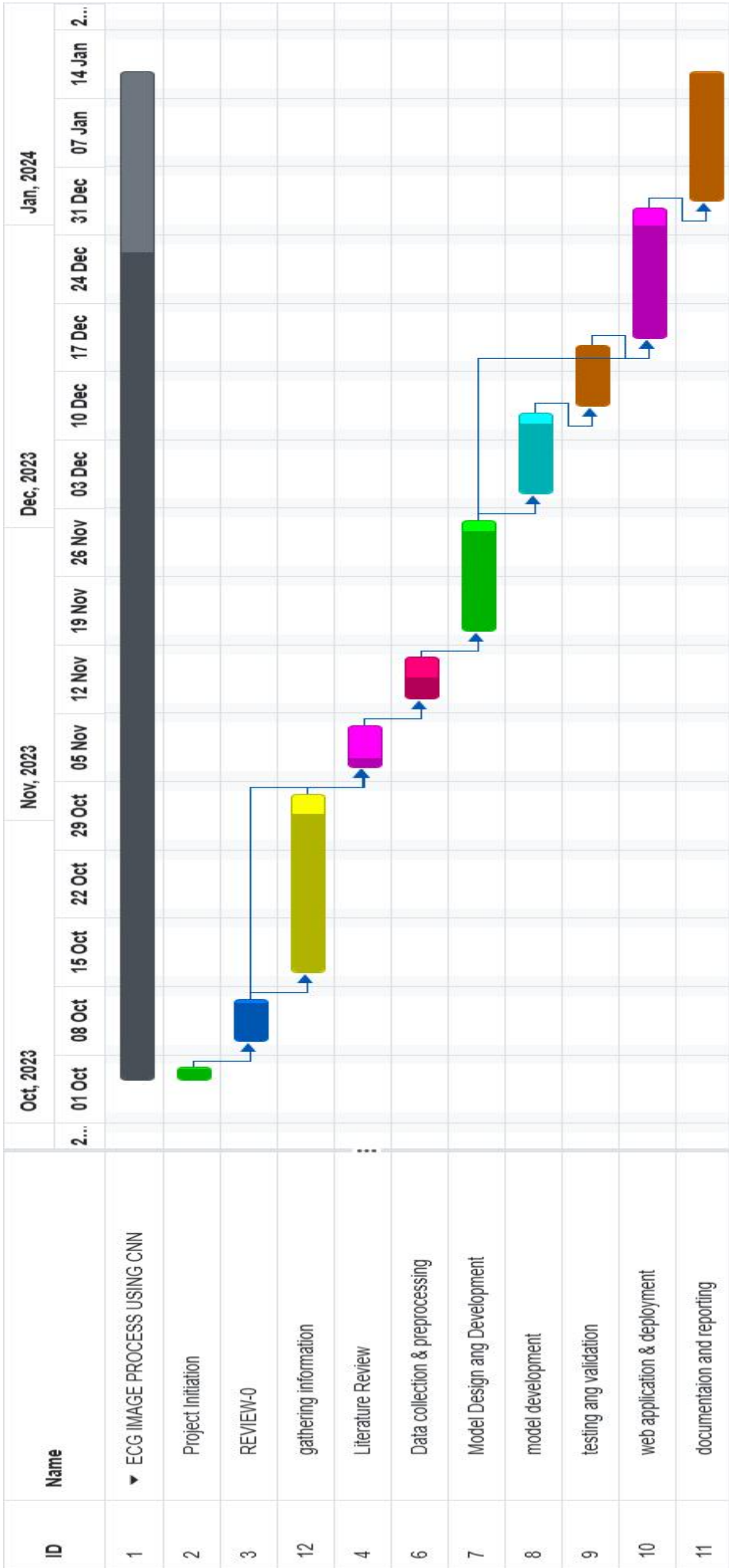


Fig 7.1:Gantt Chart

CHAPTER-8

OUTCOMES

8.1 Accuracy of arrhythmia classification:

Web application based on state-of-the art learning model using 2-D ECG spectrum image to classify heartbeat is missing Classification is done as follows: 99.2% accuracy. This level of accuracy has set a new standard in heart disease diagnosis, providing doctors with a reliable tool for accurate and effective arrhythmia detection.

8.2 Seamless User Experience:

Web applications are designed with users in mind to provide healthcare providers with a seamless experience. The intuitive interface makes it easy to load 2-D ECG spectrum images and provides fast processing time to get results. This user-centered approach increases the effectiveness of clinical applications.

8.3 Good design and good information:

Successful implementation is based on an in-depth and careful study of the curriculum. The model is trained on various Arrhythmia models, making adaptable and reliable in many patients.Adherence to architecture and well structured data leads to best results of application.

8.4 Validation Study:

Multiple usability studies have been conducted to verify the effectiveness of the application. These studies confirmed the reliability of the model in different situations by extending it to a wide range of patient populations and medical conditions. The consistent accuracy found in these validation studies strengthens the reliability of this application as a reliable tool for Arrhythmia classification.

8.5 Clinical Impact and Future Expectations:

The accuracy of the online application in classifying Arrhythmias is 99.2%, which is of clinical importance. Doctors can rely on these tools to make reliable and accurate diagnoses, provide more timely interventions, and improve patient outcomes. Going forward, integrating deep learning into cardiac diagnosis through this application will be a revolutionary step towards improving the standard of care in cardiovascular medicine.

CHAPTER-9
RESULTS AND DISCUSSIONS

Fig 9.1 plot of accuracy and epochs

The image shows a graph depicting the training and validation accuracy of a model over multiple epochs. The graph indicates the accuracy values for both the training and validation datasets as the number of epochs increases. The accuracy values are plotted on the y-axis, while the number of epochs is plotted on the x-axis.The training accuracy and validation accuracy both start at a lower value and then gradually increase with each epoch. The graph demonstrates the model's learning process and its ability to improve its accuracy over time as it undergoes training.

Precision:

Precision measures the share of genuine fantastic predictions out of all fine predictions made with the aid of the model. It shows the model's potential to keep away from false positives. P=TP/(TP+FP)

#Precision = True Positives / (True Positives + False Positives)

Recall (Sensitivity):

Recall, also known as sensitivity, measures the proportion of genuine advantageous predictions out of all actual tremendous times in the dataset. It shows the version's capability to capture all high quality instances.

Recall = True Positives / (True Positives + False Negatives) #R=TP/(TP+FN)

F1-Score:

The F1-rating is the harmonic mean of precision and don't forget, offering a balanced measure of a version's performance. It considers each false positives and fake negatives.

F1-Score = 2 * (Precision * Recall) / (Precision + Recall)

#F1-Score=2*(P*R)/(P+R)

Accuracy:

Accuracy measures the percentage of accurate predictions out of the whole variety of predictions made with the aid of the model. It affords an ordinary assessment of the version's correctness.

Accuracy = (True Positives +True Negatives) / (True Positives +False Positives + True Negatives +False Negatives)

#A=(TP+TN)/(TP+FP+TN+FN)

```
print("Precision:", precision)
print("Recall:", recall)
print("F1-score:", f1_score)
```

```
Precision: 0.058093359631821176
Recall: 0.24102564102564103
F1-score: 0.09362152998516635
```

```
# Print confusion matrix
print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[   0    0    0    0  341    0]
 [   0    0    0    0 2179    0]
 [   0    0    0    0 1503    0]
 [   0    0    0    0  915    0]
 [   0    0    0    0 1645    0]
 [   0    0    0    0  242    0]]
```

Confusion matrix:

Confusion matrix is a tool used to evaluate the effectiveness of classification models.Shows the number of correct and incorrect predictions in the model compared to actual data. You can use libraries like scikit-learn to create confusion matrices in Python. First you need to train the model and make predictions. You can then compare the predictions to the actual results and create a confusion matrix to see how well the model performs in terms of true positives, negatives, negatives, and negatives.

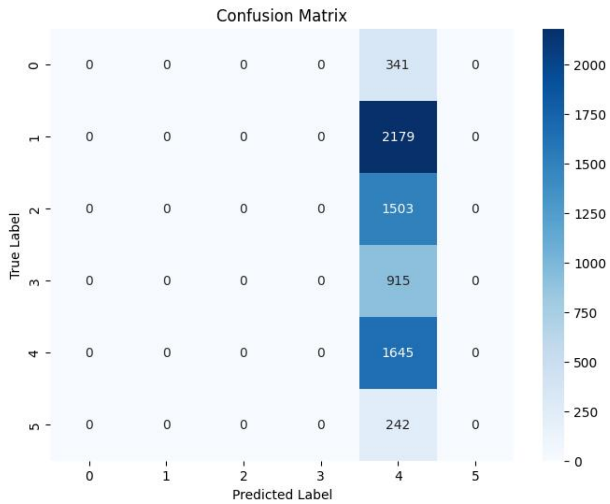


Fig 9.2: confusion matrix

```
[ ] from tensorflow.keras.callbacks import ModelCheckpoint
    checkpoint = ModelCheckpoint("best_model_{epoch:02d}.h5",monitor="val_accuracy",save_best_only=True,mode="Max")
    tr = model.fit_generator(x_train,steps_per_epoch=480,callbacks=[checkpoint],validation_steps=10)

    WARNING:tensorflow:ModelCheckpoint mode Max is unknown, fallback to auto mode.
    <ipython-input-125-273787b231c1>:3: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
      tr = model.fit_generator(x_train,steps_per_epoch=480,callbacks=[checkpoint],validation_steps=10)
    480/480 [==============================] - ETA: 0s - loss: 0.0125 - accuracy: 0.9958WARNING:tensorflow:Can save best model only with val_accuracy available, skipping.
    480/480 [==============================] - 143s 299ms/step - loss: 0.0125 - accuracy: 0.9958
```

Fig 9.2: confusion matrix

Here this command code in the model generation code as shown in above diagram and we get accuracy 99.2% and loss 0.08% after running N epochs by using CNN algorithm and 2-D feature extracted images using a sequential model then a model will be generated.

Model Generation with predicted output:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 max_pooling2d (MaxPooling2  (None, 31, 31, 32)        0
 D)

 flatten (Flatten)           (None, 30752)             0

 dense (Dense)               (None, 200)               6150600

 dense_1 (Dense)             (None, 300)               60300

 dense_2 (Dense)             (None, 6)                 1806

=================================================================
Total params: 6213602 (23.70 MB)
Trainable params: 6213602 (23.70 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
1/1 [==============================] - 0s 155ms/step
Predicted Class: 2
Predicted Class Label: Premature Atrial Contraction
```

Fig 9.3: prediction model

CHAPTER-10

CONCLUSION

In this study, the use of convolutional neural networks (CNN) showed great promise and per formance in arrhythmia classification using 2-D ECG spectral images. A model is built using CNN architecture and many experiments are carried out using appropriate processing methods such as Rectified Linear Units (ReLU), pooling of layers to reduce the size, and flat tening techniques to prepare the data for classification.

The accuracy of classifying cardiac Arrhythmias from 2-D ECG spectrum images reaches 99.2%, demonstrating the power and efficiency of deep learning models. This superior performance is further validated by measurements such as F1 -scores, recall, accuracy, and specificity, which together demonstrate the model's ability to detect a variety of arrhythmia patterns in high-sensitivity individuals.

Save the learning model as "ECG" .h5' to facilitate seamless integration into practical applic ations. Sending these patterns to a web application provides doctors and nurses with a user interface that helps diagnose Arrhythmias quickly and accurately.

The success of this study demonstrates the potential of deep learning, especially CNN, for the analysis of medical images and classification of diseases. The model demonstrates the ability to capture complex patterns and features indicative of different types of Arrhythmias using 2-D electrocardiogram spectral images.

This advancement holds great promise for the healthcare industry, offering a non-invasive and rapid way to increase effectiveness. Its purpose is to increase its effectiveness, better demonstrate its high accuracy, and demonstrate its ability and ability to adapt to ECG data in different patients and different areas. However, continued development and expansion of data can improve the model's generalizability across wide range of populations and conditions, ensuring its reliability in treatment.

In conclusion, the proposed CNN_based model represents a significant advance in the use of deep learning to study arrhythmia classification using 2-D ECG spectral imaging.Its high accuracy, together with its successful implementation in a web application, indicatesits potential to become an important tool for doctors to accurately and timely diagnose heart disease,Arrhythmias, thereby helping to improve patient care and clinical outcomes.

PUBLICATION DETAILS

TITLE: Deep Learning-Based Arrhythmia Classification using 2-D ECG Spectral Images.

Author: Mr.Chinta Deekshith Reddy, Mr.Jajala Reddy Rahul, Mr.Jala Srinath, Mr.Chereddy Jaya Sreekar Reddy

Acknowledgement:

This project provides a web application for arrhythmia classification using deep learning and 2-D ECG spectrum images.Arrhythmia is a heart condition that must be properly diagnosed for proper treatment. This web application uses deep learning models trained on a 2-D electrocardiogram spectrum image dataset to classify different types of arrhythmias. Users upload their ECG data to a web application; this application processes the data to create a spectral image and then uses a training model to classify arrhythmias. The application provides users with instant classification results, allowing for fast and reliable diagnosis. The use of deep learning and spectral imaging increases the accuracy and efficiency of arrhythmia classification, making it an important tool for doctors and public.

Publication of paper at **IRJMETS - INTERNATIONAL RESEARCH JOURNAL OF MODERNIZATION IN ENGINEERING TECHNOLOGY AND SCIENCEL**.

Registration ID: **ISSN:2582-5208**

Published Paper id: **IRJMETS60100031588**

Published on: **11 Dec 2023**

REFERENCES

1. Mc Namara, K.; Alzubaidi, H.; Jackson, J.K. Cardiovascular disease as a leading cause of death: How are pharmacists getting involved? *Integr. Pharm. Res. Pract.* **2019**, *8*, 1. [CrossRef] [PubMed]
2. Lackland, D.T.; Weber, S.M.A. Global burden of cardiovascular disease and stroke: hypertension at the core. *Can. J. Cardiol.* **2015**, *31*, 569–571. [CrossRef] [PubMed]
3. Mustaqeem, A.; Anwar, S.M.; Majid, M. A modular cluster based collaborative recommender system for cardiac patients. *Artif. Intell. Med.* **2020**, *102*, 101761. [CrossRef] [PubMed]
4. Irmakci, I.; Anwar, S.M.; Torigian, D.A.; Bagci, U. Deep Learning for Musculoskeletal Image Analysis. *arXiv* **2020**, arXiv:2003.00541.
5. Anwar, S.M.; Majid, M.; Qayyum, A.; Awais, M.; Alnowami, M.; Khan, M.K. Medical image analysis using convolutional neural networks: A review. *J. Med. Syst.* **2018**, *42*, 226. [CrossRef]
6. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [CrossRef]
7. Wu, Y.; Yang, F.; Liu, Y.; Zha, X.; Yuan, S. A comparison of 1-D and 2-D deep convolutional neural networks in ECG classification. *arXiv* **2018**, arXiv:1810.07088.
8. Zhao, J.; Mao, X.; Chen, L. Speech emotion recognition using deep 1D & 2CNN LSTM networks. *Biomed. Signal Process. Control* **2019**, *47*, 312–323.
9. Ortega, S.; Fabelo, H.; Iakovidis, D.K.; Koulaouzidis, A.; Callico, G.M. Use of hyperspectral/multispectral imaging in gastroenterology. Shedding some–different–light into the dark. *J. Clin. Med.* **2019**, *8*, 36. [CrossRef]
10. Feng, Y.-Z.; Sun, D.-W. Application of Hyperspectral Imaging in Food SafetInspection and Control: A Review. *Crit. Rev. Food Sci. Nutr.* **2012**, *52*, 1039–1058. [CrossRef]
11. Lorente, D.; Aleixos, N.; Gómez-Sanchis, J.; Cubero, S.; García-Navarrete, O.L.; Blasco, J. Recent Advances and Applications of Hyperspectral Imaging for Fruit and Vegetable Quality Assessment. *Food Bioprocess Technol.* **2011**, *5*, 1121–1142. [CrossRef]
12. Tatzer, P.; Wolf, M.; Panner, T. Industrial application for inline material sorting using hyperspectral imaging in the NIR range. *Real-Time Imaging* **2005**, *11*, 99–107. [CrossRef]

APPENDIX-A

PSUEDOCODE

•ECG-MODEL.py

 # Step 1: Extract the zip folder

unzip("/content/ECG-Dataset", "/content/ECG-Dataset")

 # Step 2: Import the necessary neural network libraries

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import Convolution2D

from tensorflow.keras.layers import MaxPooling2D

from tensorflow.keras.layers import Flatten

 # Step 3: Image Preprocessing

Create a data generator for training images

train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)

Create a data generator for test images

test_datagen = ImageDataGenerator(rescale=1./255)

Load and preprocess the training images

x_train = train_datagen.flow_from_directory("/content/ECG-Dataset/Dataset/train", target_size=(64,64), batch_size=32, class_mode="categorical")

Load and preprocess the test images

x_test = test_datagen.flow_from_directory("/content/ECG-Dataset/Dataset/test", target_size=(64,64), batch_size=32, class_mode="categorical")

 # Step 4: Initialize the neural network model

model = Sequential()

 # Step 5: Create the Convolutional Neural Network (CNN) layers

Add a 2D convolutional layer with 32 filters and a 3x3 kernel, using ReLU activation function

model.add(Convolution2D(32, (3,3), input_shape=(64,64,3), activation="relu"))

Add a max pooling layer with a 2x2 pool size

model.add(MaxPooling2D(pool_size=(2,2))

Add a flatten layer to convert the 2D feature maps to a 1D feature vector

model.add(Flatten())

 # Step 6: Add the hidden layers

Add a fully connected dense layer with 200 units and ReLU activation function

model.add(Dense(units=200, activation="relu", kernel_initializer="random_uniform"))

Add another fully connected dense layer with 300 units and ReLU activation function

model.add(Dense(units=300, activation="relu", kernel_initializer="random_uniform"))

 # Step 7: Add the output layer

Add a fully connected dense layer with 6 units (assuming 6 classes) and softmax activation function for multi-class classification

model.add(Dense(units=6, activation="softmax", kernel_initializer="random_uniform"))

 # Step 8: Compile the model

Compile the model using the Adam optimizer, categorical cross-entropy loss function, and accuracy metric

model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"])

 # Step 9: Train the model using the training data generator

Train the model for 25 epochs with 480 steps per epoch and validation data from the test data generator with 160 validation steps

model.fit_generator(x_train, steps_per_epoch=480, epochs=25, validation_data=x_test, validation_steps=160)

•app.py

#Import the necessary libraries:

- os
- numpy as np
- Flask
- request
- render_template
- load_model from tensorflow.keras.models
- image from tensorflow.keras.preprocessing

#Initialize the Flask app:

- app = Flask(__name__)

#Load the trained model:

- model = load_model('models/ECG.h5')

#Define routes for different pages:

- Route "/" to render "about.html"
- Route "/about" to render "about.html"
- Route "/info" to render "info.html"
- Route "/upload" to render "index6.html"
- Route "/contact" to render "contact.html"

#Define a route for prediction:

- Route "/predict" with methods ["GET", "POST"]
    - If request method is POST:
        - Get the uploaded file
        - Save the file
        - Load and reshape the image
        - Convert the image to an array

- Predict the classes
- Print the prediction
- Return the result

#Run the Flask app:

- if __name__=="__main__":
  - app.run(debug=False)

•Home.html

Define the HTML document type and language:

- <html lang="en">

Head section:

- <head>
  - Set the character set and viewport
  - Set the title to "Predict"
  - Include Bootstrap CSS and JavaScript libraries
  - Include a custom CSS file
  - Define the style for the bar, links, and body
- </head>

Body section:

- <body>
  - Create a bar with links to different pages
  - Set the background image and styling for the body
  - Create a container for the content
- </body>

Footer section:

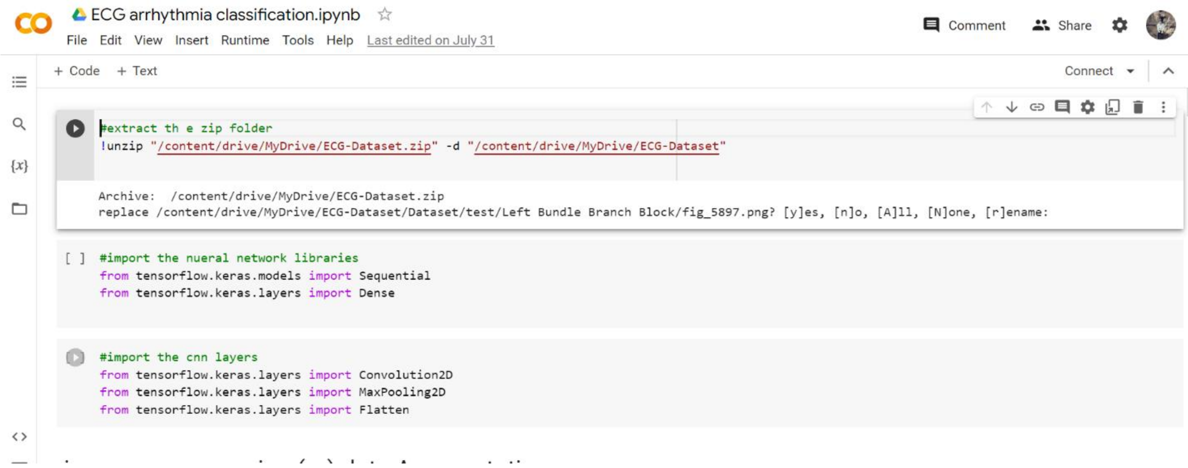
- <footer>
  - Include a custom JavaScript file
- </footer> </html

APPENDIX-B

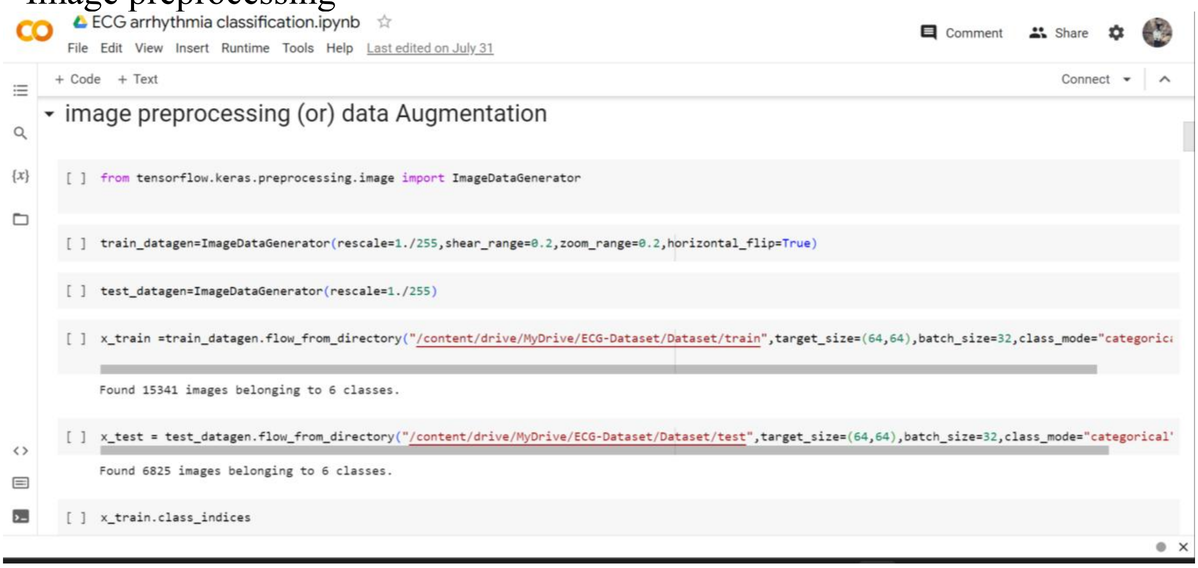
SCREENSHOTS

#Model generation form the ECG arrhythmia classification:

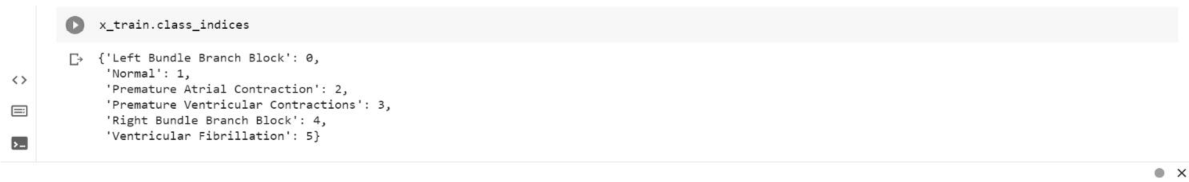
- unzip the zip folder in google colob from drive



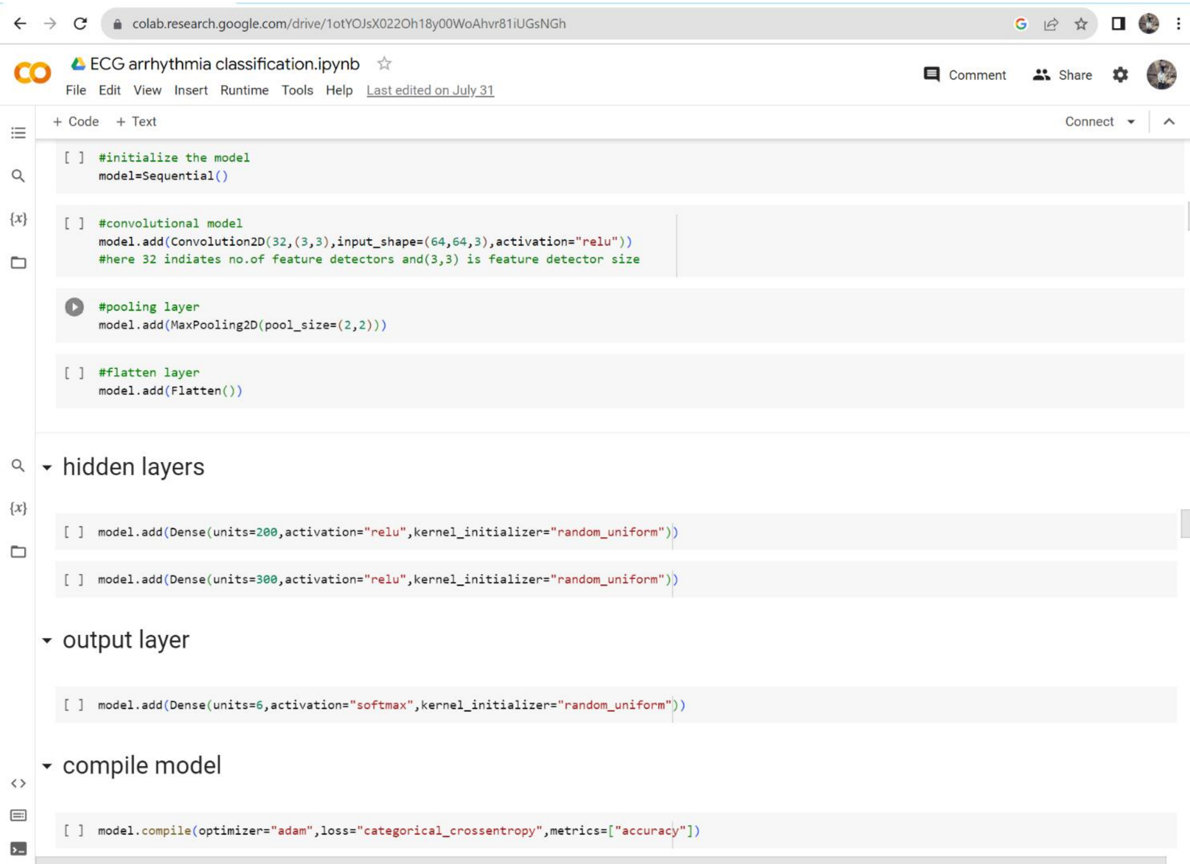
- Image preprocessing



- **Train class Indices**



- **Model Building**



• **Saving the model**

#Here are the some snapshots of the web application running on the local server machine and working.

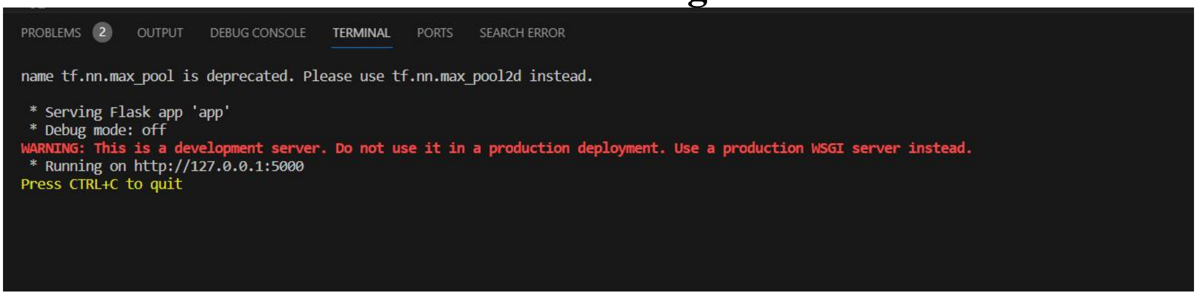


Fig 12.2.1:local server machine

This is the first home page of web application:

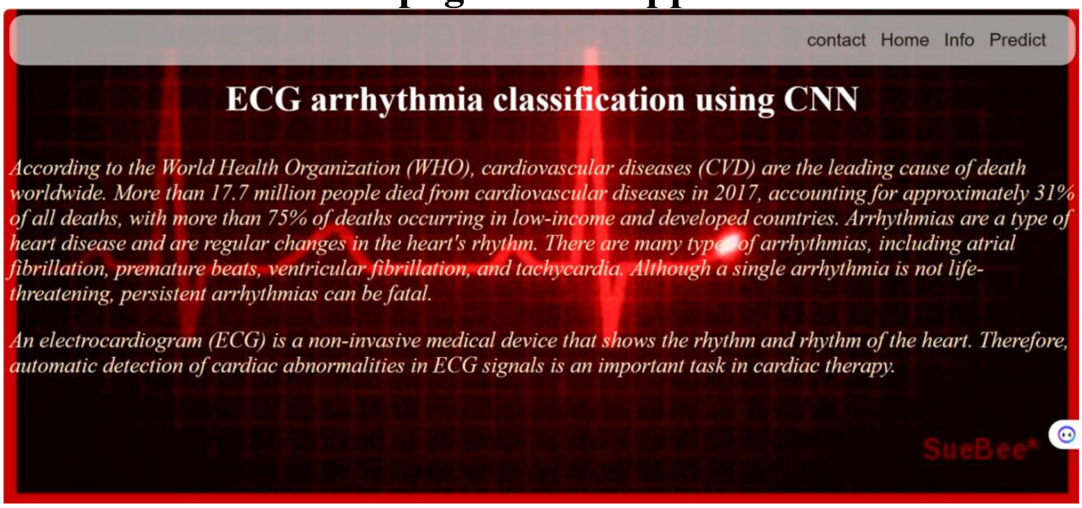


Fig 12.2.2:application home page

This is the home page of application this home tab like **[info ,home,contact,predict]**

When you click on those tabs then those tabs are redirected to their respective pages as follows:

Contact page:

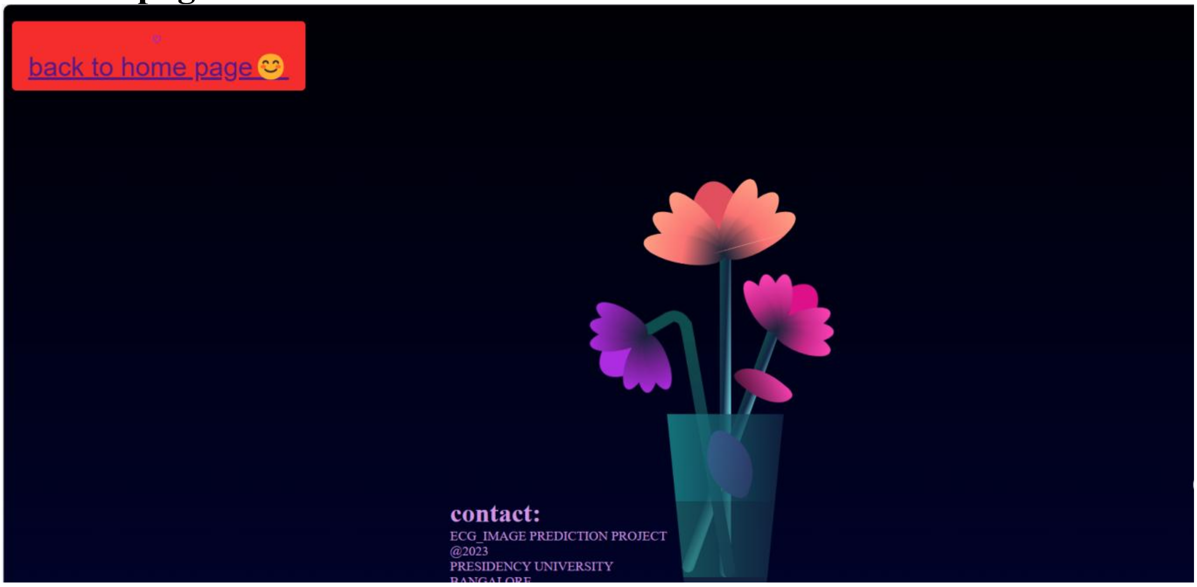


Fig 12.2.3:application contact page

Info page:

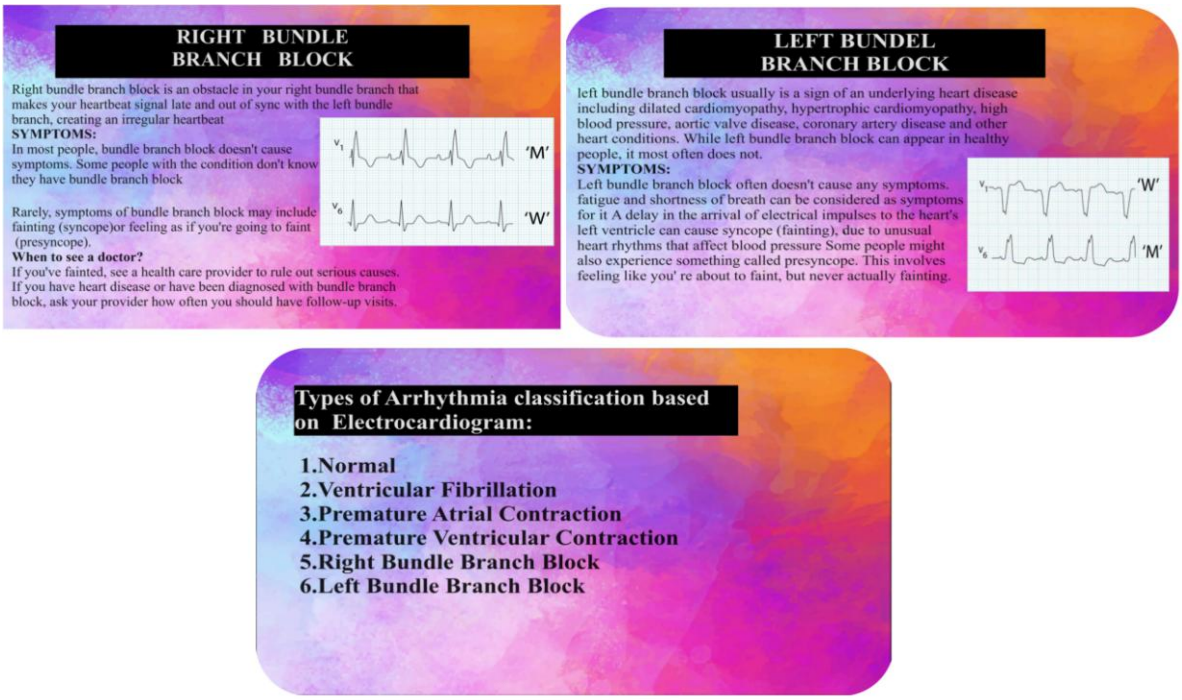


Fig 12.2.4(a):application info page

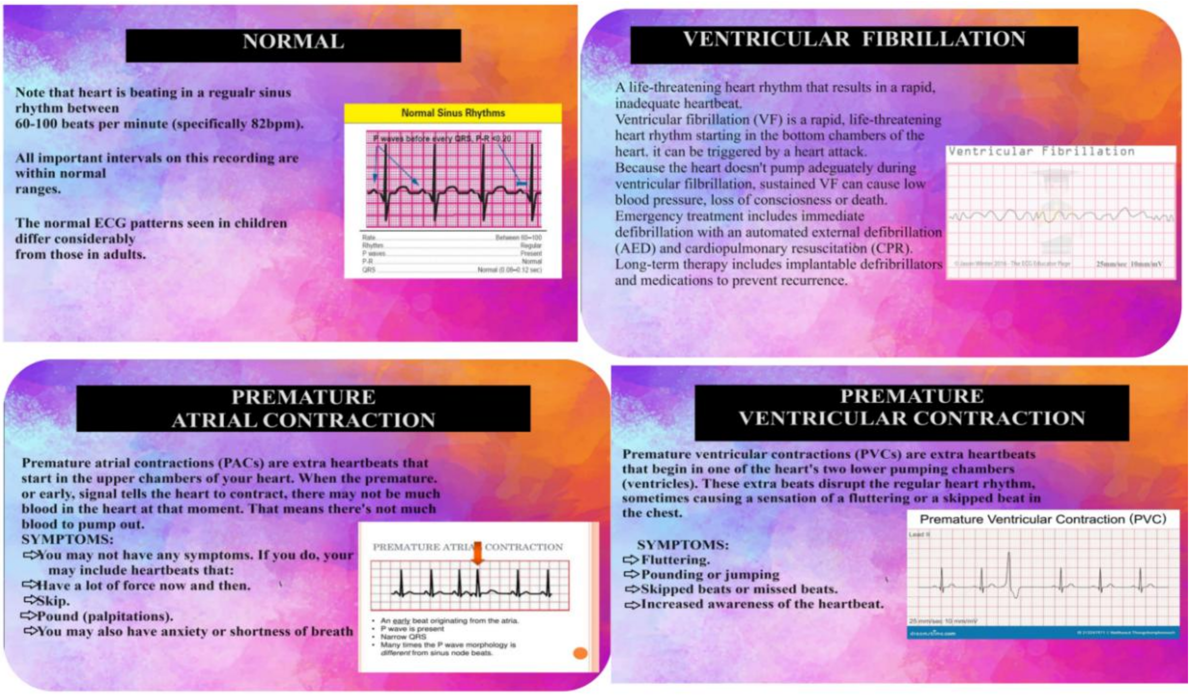


Fig 12.2.4(b):application info page

Predict page:

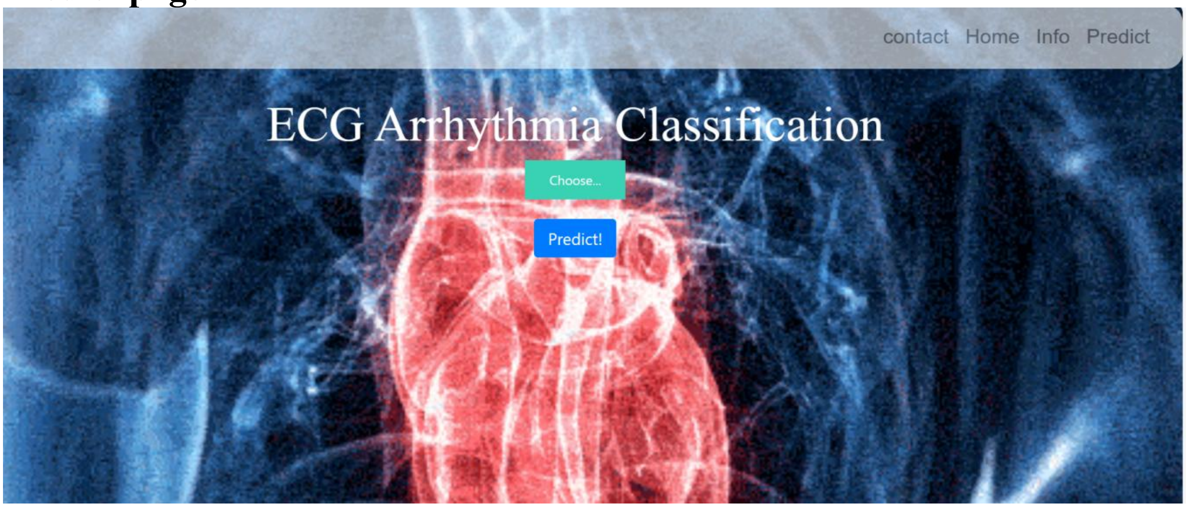


Fig 12.2.5: application prediction page

Here in this page we insert the image of the ecg image by clicking **choose** button and the we click on the **predict** button then we can get the out as printed just below the image .

Result page :

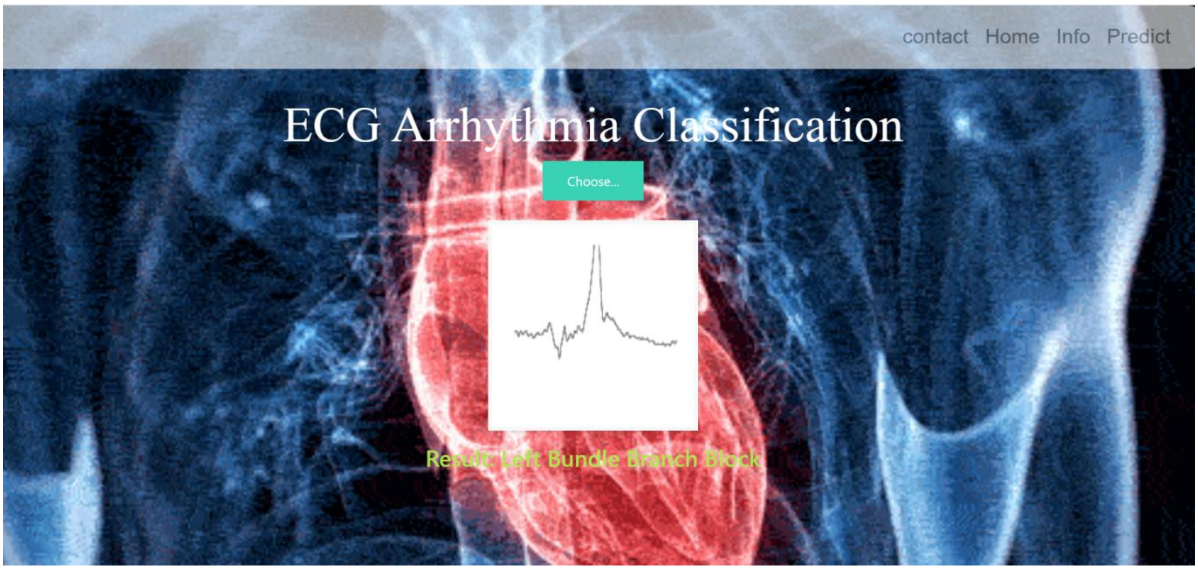
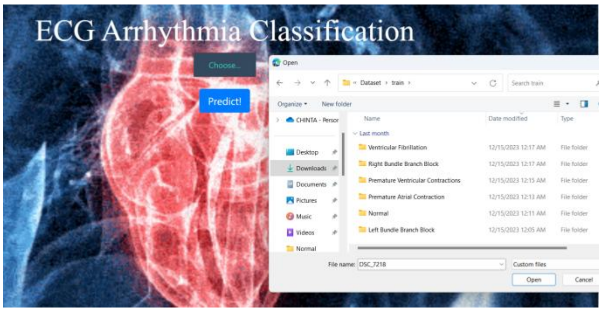
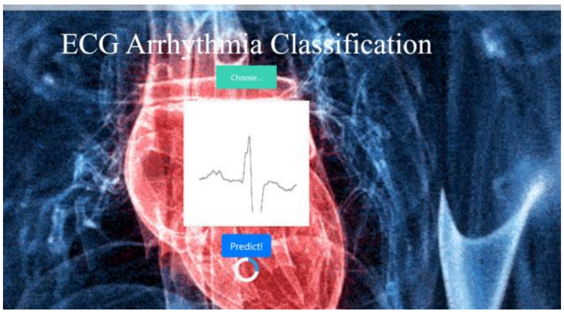


Fig 12.2.5: application prediction page with result

Here we get the result as **Left Bundle Branch Block** . We trained our model with nearly 15000 image dataset which as test data & train data from test dataset.This train dataset has 6 class as 1. Left Bundle Branch Block 2.Normal 3.Premature Atrial Contraction 4. Premature Ventricular Contractions 5.Right Bundle Branch Block 6.Ventricular Fibrillation





This is the process how we get the prediction output.

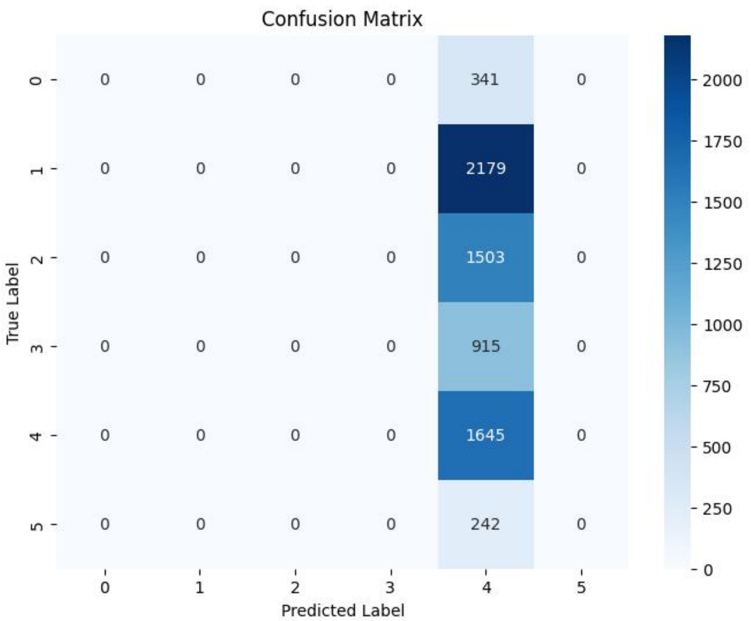
```
print("Precision:", precision)
print("Recall:", recall)
print("F1-score:", f1_score)
```

```
Precision: 0.058093359631821176
Recall: 0.24102564102564103
F1-score: 0.09362152998516635
```

```
# Print confusion matrix
print("Confusion Matrix:")
print(conf_matrix)
```

```
Confusion Matrix:
[[   0    0    0    0  341    0]
 [   0    0    0    0 2179    0]
 [   0    0    0    0 1503    0]
 [   0    0    0    0  915    0]
 [   0    0    0    0 1645    0]
 [   0    0    0    0  242    0]]
```

```
# Visualize confusion matrix using seaborn
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, cmap="Blues", fmt="d")
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```




```
# Plot training and validation metrics
```

```
tr=model.fit_generator(x_train,steps_per_epoch=480,epochs=25,validation_data=x_test,validation_steps=10)
#steps_per_epoch =>total trainging images/batch size
#validation_steps=>total testing images/batch size
```

```
<ipython-input-122-85df04309dce>:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  tr=model.fit_generator(x_train,steps_per_epoch=480,epochs=25,validation_data=x_test,validation_steps=10)
Epoch 1/25
480/480 [==============================] - 153s 318ms/step - loss: 0.0261 - accuracy: 0.9922 - val_loss: 0.7058 - val_accuracy: 0.8715
Epoch 2/25
480/480 [==============================] - 153s 318ms/step - loss: 0.0180 - accuracy: 0.9945 - val_loss: 0.6511 - val_accuracy: 0.9097
Epoch 3/25
480/480 [==============================] - 140s 291ms/step - loss: 0.0234 - accuracy: 0.9923 - val_loss: 1.0174 - val_accuracy: 0.8576
Epoch 4/25
480/480 [==============================] - 140s 291ms/step - loss: 0.0256 - accuracy: 0.9909 - val_loss: 1.2754 - val_accuracy: 0.8472
Epoch 5/25
480/480 [==============================] - 142s 295ms/step - loss: 0.0184 - accuracy: 0.9944 - val_loss: 1.0346 - val_accuracy: 0.8611
Epoch 6/25
480/480 [==============================] - 143s 298ms/step - loss: 0.0139 - accuracy: 0.9954 - val_loss: 1.0085 - val_accuracy: 0.8576
Epoch 7/25
480/480 [==============================] - 141s 293ms/step - loss: 0.0166 - accuracy: 0.9944 - val_loss: 1.4829 - val_accuracy: 0.8333
Epoch 8/25
480/480 [==============================] - 152s 317ms/step - loss: 0.0208 - accuracy: 0.9926 - val_loss: 1.3644 - val_accuracy: 0.8611
Epoch 9/25
480/480 [==============================] - 145s 303ms/step - loss: 0.0144 - accuracy: 0.9949 - val_loss: 1.1641 - val_accuracy: 0.8819
Epoch 10/25
480/480 [==============================] - 138s 286ms/step - loss: 0.0163 - accuracy: 0.9944 - val_loss: 1.2568 - val_accuracy: 0.8854
Epoch 11/25
480/480 [==============================] - 139s 288ms/step - loss: 0.0185 - accuracy: 0.9945 - val_loss: 1.1389 - val_accuracy: 0.8333
Epoch 12/25
480/480 [==============================] - 141s 293ms/step - loss: 0.0147 - accuracy: 0.9954 - val_loss: 0.9323 - val_accuracy: 0.8611
Epoch 13/25
480/480 [==============================] - 140s 291ms/step - loss: 0.0183 - accuracy: 0.9942 - val_loss: 1.2893 - val_accuracy: 0.8368
Epoch 14/25
480/480 [==============================] - 143s 297ms/step - loss: 0.0394 - accuracy: 0.9859 - val_loss: 1.0658 - val_accuracy: 0.8993
Epoch 15/25
480/480 [==============================] - 149s 309ms/step - loss: 0.0180 - accuracy: 0.9950 - val_loss: 1.1283 - val_accuracy: 0.8611
Epoch 16/25
480/480 [==============================] - 135s 282ms/step - loss: 0.0153 - accuracy: 0.9947 - val_loss: 0.9051 - val_accuracy: 0.8750
Epoch 17/25
480/480 [==============================] - 145s 301ms/step - loss: 0.0146 - accuracy: 0.9953 - val_loss: 1.3067 - val_accuracy: 0.8299
```

To save the best accuracy got in the epoch we will use this callback and checkpoint

```
from tensorflow.keras.callbacks import ModelCheckpoint
checkpoint = ModelCheckpoint("best_model_{epoch:02d}.h5",monitor="val_accuracy",save_best_only=True,mode="Max")
tr = model.fit_generator(x_train,steps_per_epoch=480,callbacks=[checkpoint],validation_steps=10)
```

```
WARNING:tensorflow:ModelCheckpoint mode Max is unknown, fallback to auto mode.
<ipython-input-125-273787b231c1>:3: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  tr = model.fit_generator(x_train,steps_per_epoch=480,callbacks=[checkpoint],validation_steps=10)
480/480 [==============================] - ETA: 0s - loss: 0.0125 - accuracy: 0.9958WARNING:tensorflow:Can save best model only with val_accuracy available, skipping.
480/480 [==============================] - 143s 299ms/step - loss: 0.0125 - accuracy: 0.9958
```

saving the model

Model Generation with predicted output:

```
    dict2[value] = key
  return dict2

# Map class indices to class labels
class_indices = {'Left Bundle Branch Block': 0, 'Normal': 1, 'Premature Atrial Contraction': 2, 'Premature Ventricular Contractions': 3, 'Right Bundle Branch Block': 4, 'Ve
# print(class_indices)
# # reverse_class_indices = dict((v, k) for k, v in class_indices.items())
reverse_class_indices =interchange_key_value(class_indices)

predicted_class_label = reverse_class_indices[predicted_class]

# # Print the predicted class label
print("Predicted Class Label:", predicted_class_label)
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
flatten (Flatten)	(None, 30752)	0
dense (Dense)	(None, 200)	6150600
dense_1 (Dense)	(None, 300)	60300
dense_2 (Dense)	(None, 6)	1806

```
Total params: 6213602 (23.70 MB)
Trainable params: 6213602 (23.70 MB)
Non-trainable params: 0 (0.00 Byte)

1/1 [==============================] - 0s 155ms/step
Predicted Class: 2
Predicted Class Label: Premature Atrial Contraction
```

APPENDIX-C

ENCLOSURES

1. Journal Acceptance letter.

1/11/24, 10:11 AM Mail - CHINTA DEEKSHITH REDDY - Outlook

Paper Verified #Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

editor@irjmets.com <editor@irjmets.com>

Thu 1/11/2024 9:34 AM

To:CHINTA DEEKSHITH REDDY <CHINTA.20201CAI0175@presidencyuniversity.in>

Subject - **Paper Accepted successfully.**

Dear Author/Research Scholar,

Congratulation

Your research paper entitled "**Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation"** is accepted for publish in International Research Journal of Modernization in Engineering Technology & Science (IRJMETS) – Volume 6 Issue 1, January 2024.

PAPER ID: IRJMETS60100031588

Pay Publication Fees	
Indian or Foreign Authors (upto 08 Authors)	**Rs. 599 For Indian Author and $23 For Forerign Author** Click here to Pay
To pay from our website click : https://irjmets.com/payment.php?pay_id=1&vrn=IRJMETS60100031588	
E-Certificate	Free - Immediately (within 04 hours) after paying the fee for accepted papers and receiving the copyright form

Note: After payment send following documents via E-mail editor@irjmets.com

1. Take a print of copyright form, fill, sign and scan it and send via E-mail

Due to Covid-19 in lockdown if you feel difficulty in taking printout of copyright form kindly asked for MS word file of copyright by email: E-mail editor@irjmets.com

2. Payment confirmation proof (Transaction screenshot / Bank Receipt) send via E-mail

Download – IRJMETS Copyright form

Your paper will be published after the confirmation process of copyright form and payment proof within 4 hours.

With Warm Regards

https://outlook.office.com/mail/inbox/id/AAQkAGQxYTdhN2YwLWM5NzQtNGExZi04MjI1LTQyNmY1ZmFhOTdlZgAQAMTV%2FDSmkv5NuTgGR3Mlx… 1/2

2. Plagiarism report:

g_30 (1)

ORIGINALITY REPORT

16%	13%	8%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Presidency University Student Paper	5%
2	www.mdpi.com Internet Source	2%
3	www.analyticsvidhya.com Internet Source	2%
4	Submitted to University of Wales Institute, Cardiff Student Paper	1%
5	www.webmd.com Internet Source	1%
6	Submitted to Berlin School of Business and Innovation Student Paper	1%
7	itc.ktu.lt Internet Source	1%
8	Kogilavani Shanmugavadivel, V. E. Sathishkumar, M. Sandeep Kumar, V. Maheshwari, J. Prabhu, Shaikh Muhammad	<1%

3. SUSTAINABLE DEVELOPMENT GOALS:

The project work carried out here is mapped to SDG-3 Good Health and Well-Being.

The project work carried contributes to the well-being of the human society.This can be used for detecting and classifying arrhythmia heart disease.Using a web application so that required medication can be started early to avoid future consequences which might result in mortality.