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

**A PROJECT REPORT**

***Submitted by***

**TEAM ID : PNT2022TMID48615**
**ANANDAKUMAR .A**
**PRAVEEN .T**
**HARI DEEVAGAN .M**
**BRAMMA .S**

***in partial fulfillment for the award of the degree***
***of***
**BACHELOR OF ENGINEERING**
***in***
**COMPUTER SCIENCE AND ENGINEERING**

**NPR COLLEGE OF ENGINEERING & TECHNOLOGY,**
**NATHAM - DINDIGUL.**

# TABLE OF CONTENTS

# 1. INTRODUCTION

## 1.1 Project Overview :

According to the World Health Organization (WHO), cardiovascular diseases (CVDs) are the number one cause of death today. Over 17.7 million people died from CVDs in the year 2017 all over the world which is about 31% of all deaths, and over 75% of these deaths occur in low and middle-income countries. Arrhythmia is a representative type of CVD that refers to any irregular change from the normal heart rhythms. There are several types of arrhythmia including atrial fibrillation, premature contraction, ventricular fibrillation, and tachycardia. Although a single arrhythmia heartbeat may not have a serious impact on life, continuous arrhythmia beats can result in fatal circumstances. In this project, we build an effective electrocardiogram (ECG) arrhythmia classification method using a convolutional neural network (CNN), in which we classify ECG into seven categories, one being normal and the other six being different types of arrhythmia using deep two-dimensional CNN with grayscale ECG images. We are creating a web application where the user selects the image which is to be classified. The image is fed into the model that is trained and the cited class will be displayed on the webpage.

The electrocardiogram (ECG) is one of the most extensively employed signals used in the diagnosis and prediction of cardiovascular diseases (CVDs). The ECG signals can capture the heart's rhythmic irregularities, commonly known as arrhythmias. A careful study of ECG signals is crucial for precise diagnoses of patients' acute and chronic heart conditions. In this study, we propose a two-dimensional (2-D) convolutional neural network (CNN) model for the classification of ECG signals into eight classes; namely, normal beat, premature ventricular contraction beat, paced beat, right bundle branch block beat, left bundle branch block beat, atrial premature contraction beat, ventricular flutter wave beat, and ventricular escape beat. The one-dimensional ECG time series signals are transformed into 2-D spectrograms through short-time Fourier transform. The 2-D CNN model consisting of four convolutional layers and four pooling layers is designed for extracting robust features from the input spectrograms. Our proposed methodology is evaluated on a publicly available MIT-BIH arrhythmia dataset. We achieved a state-of-the-art average classification accuracy of 99.11%, which is better than those of recently reported results in classifying similar types of arrhythmias. The performance is significant in other indices as well, including sensitivity and specificity, which indicates the success of the proposed method.

## 1.2 Purpose :

The conventional techniques might not achieve efficient results due to the inter-patient variability in ECG signals. Additionally, the efficiency and accuracy of traditional methods could be negatively affected by the increasing size of data . The techniques presented in literature have been applied to smaller datasets; however, for the purpose of generalization, the performance should be tested on larger datasets. There are methods reported that use 2-D ECG signals however, to the best of our knowledge, there are not clear details on how the 1-D ECG signal is converted to 2-D images for using 2-D CNN models. Most methods have been tested on only a few types of arrhythmia and must be evaluated on all major types of arrhythmia. It should be noted that the performance of methods developed for 1-D ECG signals can be further improved. Towards this end, the major contributions of our proposed work are:

1. Spectrograms (2-D images) are employed, which are generated from the 1-D ECG signal using STFT. In addition, data augmentation was used for the 2-D image representation of ECG signals.
2. A state-of-the-art performance was achieved in ECG arrhythmia classification by using the proposed CNN-based method with 2-D spectrograms as input.

The purpose of this project is to we detect the type of arrhythmia at earlier stage and give treatment relevant to that type . we can prevent the life of people.

# 2. LITERATURE SURVEY

## 2.1 Existing Problem :

The existing system does not classified the  ECG signals into different segmentations. It highlights various segments within the heartbeat besides displaying three waves. (P wave, QRS wave, T wave). The patterns are easily the change these waves . It leads to misclassify the arrhythmia type. Dataset which have huge images were difficult to load and process. The accuracy of the classification is poor.

The existing system classification is based on only one-dimensional image spectral analysis. So due to one-dimensional image spectral analysis signal denoising is not possible. So the classification have a high chance to mismatch.

## 2.2 References :

**1)** Anwar, S.M.; Majid, M. A modular cluster based collaborative recommender system for cardiac patients. Artif. Intell. Med. 2020, 102, 101761.

**2)** Bagci, Anwar, S.M.; Torigian, D.A. UIrmakci, I.; . Deep Learning for Musculoskeletal Image Analysis. arXiv Prepr. 2020, arXiv:2003.00541.

**3)** Chen, L.Zhao, J.; Mao, X.; Speech emotion recognition using deep 1D & 2-D CNN LSTM networks. Biomed. Signal Process. Control 2019, 47, 312–323.

**4)** D.K.;Ortega, S.; Fabelo, H.; Iakovidis,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.

**5)** Daniel,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.

**6)** Hassan, H.; Bashir, A. K.; Abbasi, R.; Ahmad, W.; Luo, B. Single image defocus estimation by modified gaussian function. Trans. Emerg. Telecommun. Technol. 2019, 30, 3611.

**7)** Jayaraman,; Arul, R.; Basheer, S.; Raja, G.;R Bashir, A.K. .; Qureshi, N.M.F. An optimal multitier resource allocation of cloud RAN in 5G using machine learning. Trans. Emerg. Telecommun. Technol. 2019, 30, 3627.

**8)** Luo, Chen, J.; Valehi, A.; Razi, A. Smart Heart Monitoring: Early Prediction of Heart Problems Through Predictive Analysis of ECG Signals. IEEE Access 2019, 7, 120831–120839.

**9)** 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.

**10)** Mohamed 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.

**11)** Mustaqeem; Yang, F.; Liu, Y.; Zha, X.; Yuan, S. A comparison of 1-D and 2-D deep convolutional neural networks in ECG classification. arXiv Prepr. 2018, arXiv:1810.07088.

**12)** Majid, M.; Alnowami, M. Arrhythmia Classification of ECG Signals Using Hybrid Features. Comput. Math. Methods Med. 2018.

**13)** Mustaqeem, A.; Anwar, S.M.; Majid, M. Multiclass classification of cardiac arrhythmia using improved feature selection and SVM invariants. Comput. Math. Methods Med. 2018, https://doi.org/10.1155/2018/7310496.

**14)** Rajpurkar, ; Yu, X.; Bashir, A. K.; Chaudhry, H. N.; Wang, D. A machine learning approach for feature selection traffic classification using security analysis. J. Supercomput. 2018, 74, 4867−4892.

**15)**P.Shafiq, M. ; Hannun, A.Y.; Haghpanahi, M.; Bourn, C.; Ng, A.Y. Cardiologist-level arrhythmia detection with convolutional neural networks. arXiv Prepr. 2017, arXiv:1707.01836.

## 2.3 Problem Statement Definition :

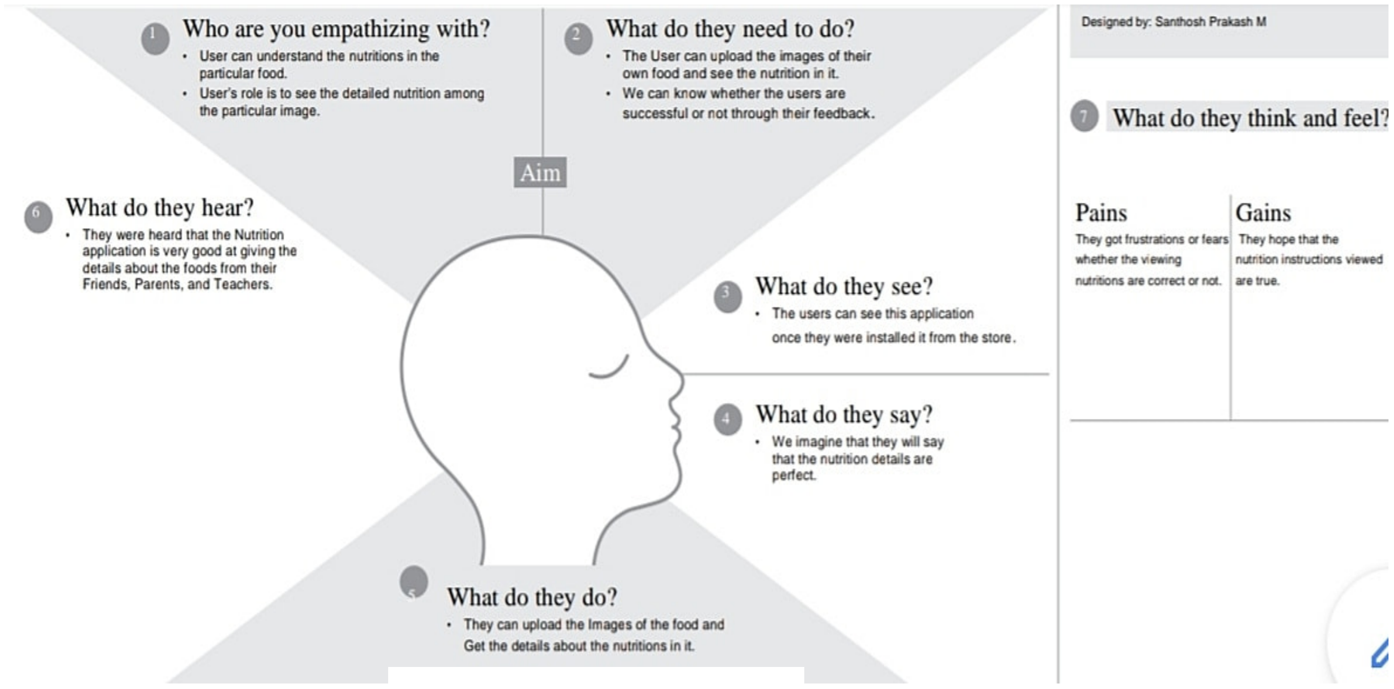
There are several types of arrhythmia including atrial fibrillation, premature contraction, ventricular fibrillation, and tachycardia. Although a single arrhythmia heartbeat may not have a serious impact on life, continuous arrhythmia beats can result in fatal circumstances. In this project, we build an effective electrocardiogram (ECG) arrhythmia classification method using a convolutional neural network (CNN), in which we classify ECG into seven categories, one being normal and the other six being different types of arrhythmia using deep two-dimensional CNN with grayscale ECG images. We are creating a web application where the user selects the image which is to be classified. The image is fed into the model that is trained and the cited class will be displayed on the webpage.

# 3. IDEATION & PROPOSED SOLUTION

## 3.1 Empathy Map Canvas

The empathy map canvas expands upon the original empathy map with a deeper emphasis on user motivations and consideration of influences that impact user decision-making. Additionally, the empathy map canvas provides basic instructions and prompts to help guide teams through its completion and streamline the process, which a basic empathy map does not do.

The Empathy Map Canvas helps teams develop deep, shared understanding and empathy for other people. People use it to help them improve customer experience, to navigate organizational politics, to design better work environments, and a host of other things.


Designed by: Santhosh Prakash M
Aim
7 What do they think and feel?
Pains
Gains

1 Who are you empathizing with?
- User can understand the nutritions in the particular food.
- User's role is to see the detailed nutrition among the particular image.

2 What do they need to do?
- The User can upload the images of their own food and see the nutrition in it.
- We can know whether the users are successful or not through their feedback.

6 What do they hear?
- They were heard that the Nutrition application is very good at giving the details about the foods from their Friends, Parents, and Teachers.

3 What do they see?
- The users can see this application once they were installed it from the store.

4 What do they say?
- We imagine that they will say that the nutrition details are perfect.

5 What do they do?
- They can upload the Images of the food and Get the details about the nutritions in it.

Pains: They got frustrations or fears whether the viewing nutritions are correct or not.

Gains: They hope that the nutrition instructions viewed are true.

Fig.3.1 Empathy Map

## 3.2 Ideation & Brainstorming :

Here the input is Raw ECG which is given to signal denoising which filters out the image so that clear image is obtained. The output received from signal denoising is given to ECG Segmentation that segments ECG into five parts. The segmented ECG images are passed onto Bispectrum Analysis. From bispectrum analysis the features of the ECG are studied at 2-D CNN.and in features extraction. Through SVM-RBF classifier the required ECG for predicting Arrhythmia is found.

Brainstorming is a group creativity technique by which efforts are made to find a conclusion for a specific problem by gathering a list of ideas spontaneously contributed by its members. In other words, brainstorming is a situation where a group of people meet to generate new ideas and solutions around a specific domain of interest by removing inhibitions. People are able to think more freely and they suggest as many spontaneous new ideas as possible. All the ideas are noted down without criticism and after the brainstorming session the ideas are evaluated.

## 3.3 Proposed Solution :

The method consists of five steps, i.e., signal pre-processing, generation of 2-D images (spectrograms), augmentation of data, extraction of features from the data (using the CNN model), and its classification based on the extracted features. The details of these steps are presented in the following subsections.

1. Pre-Processing - The three primary forms of noise in the ECG signal are power line interference, baseline drift, and electromyographic noise. The noise from the original ECG signal must be removed to ensure that a denoised ECG signal is obtained for further processing. We combined wave let based thresholding and the reconstruction algorithm of wavelet decomposition to remove noise from the original ECG signal. The wavelet thresholding was performed using, Where, w,x,y represents the wavelet coefficients, ω,x,y represents the estimated

wavelet coefficients after threshold, x represents the scale and y represents the shift, $\lambda$ represents the threshold, and $\alpha$ is a parameter whose value can be set arbitrarily. The wavelet thresholding reduced the electromyographic noise and power line noise interference. Moreover, the reconstruction algorithm of wavelet decomposition was used to remove the baseline drift noise from the noisy ECG signal .

2. Generation of 2-D Images - While 1-D CNN can be used for time series signals, the flexibility of such models is limited due to the use of 1-D kernels. On the other hand, 3-D CNNs require a large amount of training data and computational resources. In comparison, 2-D CNNs are more versatile since they use 2-D kernels and, hence, could provide representative features for time series data. Hence, for certain applications where sufficient data is available and for 1-D signals that can be represented in a 2-D format, using a 2-D CNN could be beneficial. Herein, for generating 2-D images to be used with the 2-D CNN model, the ECG signal was transformed into a 2-D representation. The 2-D time-frequency spectrograms were generated using the short-time Fourier transform. The ECG signal represents non-stationary data where the instantaneous frequency varies with time. Hence, such changes cannot be fully represented by just using information in the frequency domain. The STFT is a method derived from the discrete Fourier transform to analyze instantaneous frequency as well as the instantaneous amplitude of a localized wave with time-varying characteristics. In the analysis of a non-stationary signal, it is assumed that the signal is approximately stationary within the span of a temporal window of finite support. The 1-D ECG signals were converted into 2-D spectrogram images by applying STFT as follows, $XSTFT[m, n] = L-1 \sum k=0 \, x[k]g[k - m]e \, -j2\pi nk/L$ where L is the window length, and x[k] is the input ECG signal. The log values of XSTFT[m, n] are represented as spectrogram ($256 \times 256$) images.

3. Data Augmentation - Another significant advantage of using 2-D CNN models is the flexibility it provides in terms of data augmentation. For 1-D ECG signals, data augmentation could change the meaning of the data and hence is not beneficial. However, with 2-D spectrograms, the CNN model can learn the data variations, and augmentation helps in increasing the amount of data available for training. The ECG data is highly imbalanced, where

most of the instances represent the normal class. In this scenario, data augmentation can help when those classes that are underrepresented are augmented. For arrhythmia classification using ECG signals, augmenting training data manually could degrade the performance. Moreover, classification algorithms such as SVM, fast Fourier neural network, and tree-based algorithms, assume that the classification of a single image based representation of an ECG signal is always the same . The proposed CNN model works on 2-D images of ECG signals as input data, which allows changing the image size with operations such as cropping. Such augmentation methods would add to the training data and hence would allow better training of the CNN model. Another important issue that arises when using small data with CNN based architectures is overfitting. Data augmentation is a way to deal with overfitting and allows better training of a CNN model. For imbalanced data, data augmentation can help in maintaining a balance between different classes. We have used the cropping method for the augmentation of seven classes of ECG beats; namely, premature ventricular contraction beat (PVC), paced beat (PAB), right bundle branch block beat (RBB), left bundle branch block beat (LBB), atrial premature contraction beat (APC), ventricular flutter wave (VFW), and ventricular escape beat (VEB). These are common types of cardiac arrhythmias and are considered in studies we have used for comparison (refer to the Discussion section). While other methods of augmentation are used, such as warping in image processing applications, the aim here is to augment classes that are under-represented. Towards this end, eight different cropping operations (left top, center top, right top, left center, center, right center, left bottom, center bottom, right bottom) were applied. As a result of cropping, we obtain multiple ECG spectrograms of reduced size ($200 \times 200$), which are then resized to $256 \times 256$ images (using linear interpolation) before being fed into the CNN. This resulted in an eight times increase in the training data, which benefited the training process.

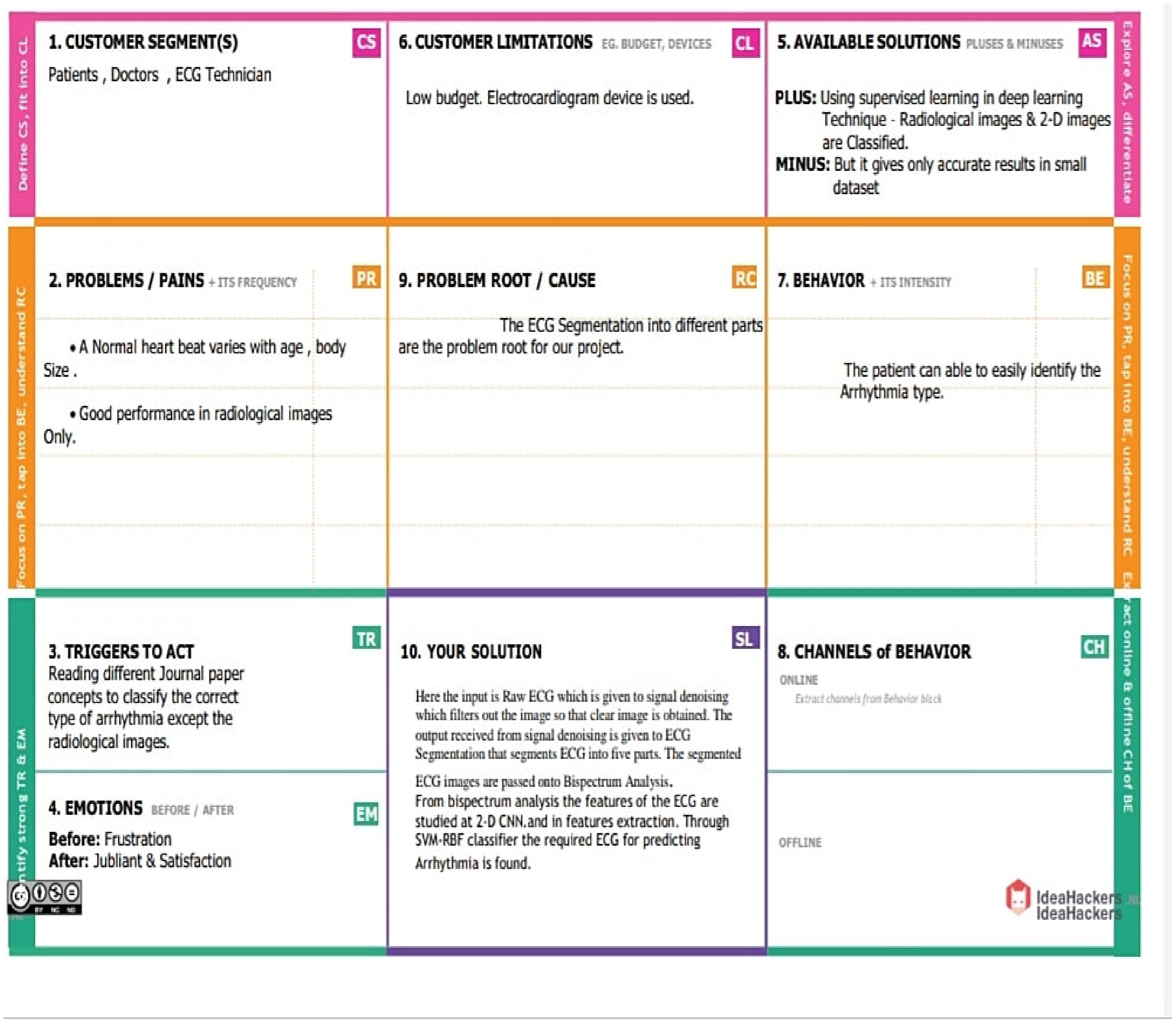
4. Deep Neural Network - In this study, a CNN-based model is proposed for an automatic classification of arrhythmia using the ECG signal in a supervised manner. The ECG data used in the study have corresponding labels (ground truth) identifying the type of arrhythmia present. These labels were assigned by expert cardiologists and are used for supervised training. For each heartbeat segment, the arrhythmia class label was transferred to the corresponding spectrogram image representation. The first CNN-based algorithm, introduced in 1989 , was developed and

used for the recognition of handwritten zip codes. Since then, multiple CNN models have been proposed for the classification of images, among which AlexNet has achieved significant performance for a variety of images. The existing neural networks with the feed-forward process for the automatic classification of the 2-D image was not feasible since these methods do not take into account the local spatial information. However, with the development of CNN architectures and using nonlinear filters, spatially adjacent pixels can be correlated to extract local features from the 2-D image. In the 2-D convolution algorithm, the downsampling layer is highly desirable for extracting and filtering the spatial vicinity of the 2-D ECG images. For these reasons, the ECG signal was transformed into a 2-D representation, and a 2-D CNN algorithm was used for classification. Consequently, high accuracy was obtained in the automatic taxonomy of the ECG waves. The details of the proposed CNN model is presented in Section.

## 3.4 Problem Solution Fit

The Problem-Solution Fit Canvas is a template to help identify solutions with higher chances of solution adoption, reduce time spent on testing and get a better overview of the current situation.

The Problem-Solution Fit is an important step towards the Product-Market Fit, but often an underestimated one.



Define CS, fit into CL

**1. CUSTOMER SEGMENT(S)** CS

Patients , Doctors , ECG Technician

**6. CUSTOMER LIMITATIONS** EG. BUDGET, DEVICES CL

Low budget. Electrocardiogram device is used.

**5. AVAILABLE SOLUTIONS** PLUSES & MINUSES AS

**PLUS:** Using supervised learning in deep learning Technique - Radiological images & 2-D images are Classified.

**MINUS:** But it gives only accurate results in small dataset

Explore AS, differentiate

Focus on PR, tap into BE, understand RC

**2. PROBLEMS / PAINS** + ITS FREQUENCY PR

- A Normal heart beat varies with age , body Size .
- Good performance in radiological images Only.

**9. PROBLEM ROOT / CAUSE** RC

The ECG Segmentation into different parts are the problem root for our project.

**7. BEHAVIOR** + ITS INTENSITY BE

The patient can able to easily identify the Arrhythmia type.

Focus on PR, tap into BE, understand RC

ntify strong TR & EM

**3. TRIGGERS TO ACT** TR

Reading different Journal paper concepts to classify the correct type of arrhythmia except the radiological images.

**4. EMOTIONS** BEFORE / AFTER EM

**Before:** Frustration

**After:** Jubliant & Satisfaction

**10. YOUR SOLUTION** SL

Here the input is Raw ECG which is given to signal denoising which filters out the image so that clear image is obtained. The output received from signal denoising is given to ECG Segmentation that segments ECG into five parts. The segmented ECG images are passed onto Bispectrum Analysis.
From bispectrum analysis the features of the ECG are studied at 2-D CNN.and in features extraction. Through SVM-RBF classifier the required ECG for predicting Arrhythmia is found.

**8. CHANNELS of BEHAVIOR** CH

ONLINE

*Extract channels from Behavior block*

OFFLINE

Extract online & offline CH of BE

IdeaHackers

Fig.3.2 Problem Solution Fit

# 4. REQUIREMENT ANALYSIS

## 4.1 Functional Requirements :

A functional requirement defines a system or its components. Functional requirements **may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish**. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in use cases.

Following are the functional requirements of the proposed solution.

| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story/ Sub-Task)** |
|---|---|---|
| FR-1 | Tester feed image | Done through mail |
| FR-2 | User Confirmation | Confirmation via Email |

## 4.2 Non - Functional Requirements :

A Non - Functional requirement defines the quality attribute of a software system.These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called as non - behavioral requirements. They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability

- Flexibility

Following are the non-functional requirements of the proposed solution.

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | We did notspecify parts of the systemfunctionality, only how that functionality is to be perceived by user. |
| NFR-2 | **Security** | The system must able to accommodate larger volumes over time.The security on their databasesmay include firewalls to prevent unauthorized access. |
| NFR-3 | **Reliability** | Highly reliable functions with same or similar efficiency after extensive use. We can assess adevice reliability: i)Percentage of probability offailure. Ii) Number of critical failures |
| NFR-4 | **Performance** | Itdefines attribute of software system. |
| NFR-5 | **Availability** | It describes how the systemis accessible to a user ata given pointin time. |
| NFR-6 | **Scalability** | It is the ability to appropriately handle increasingworkloads. |

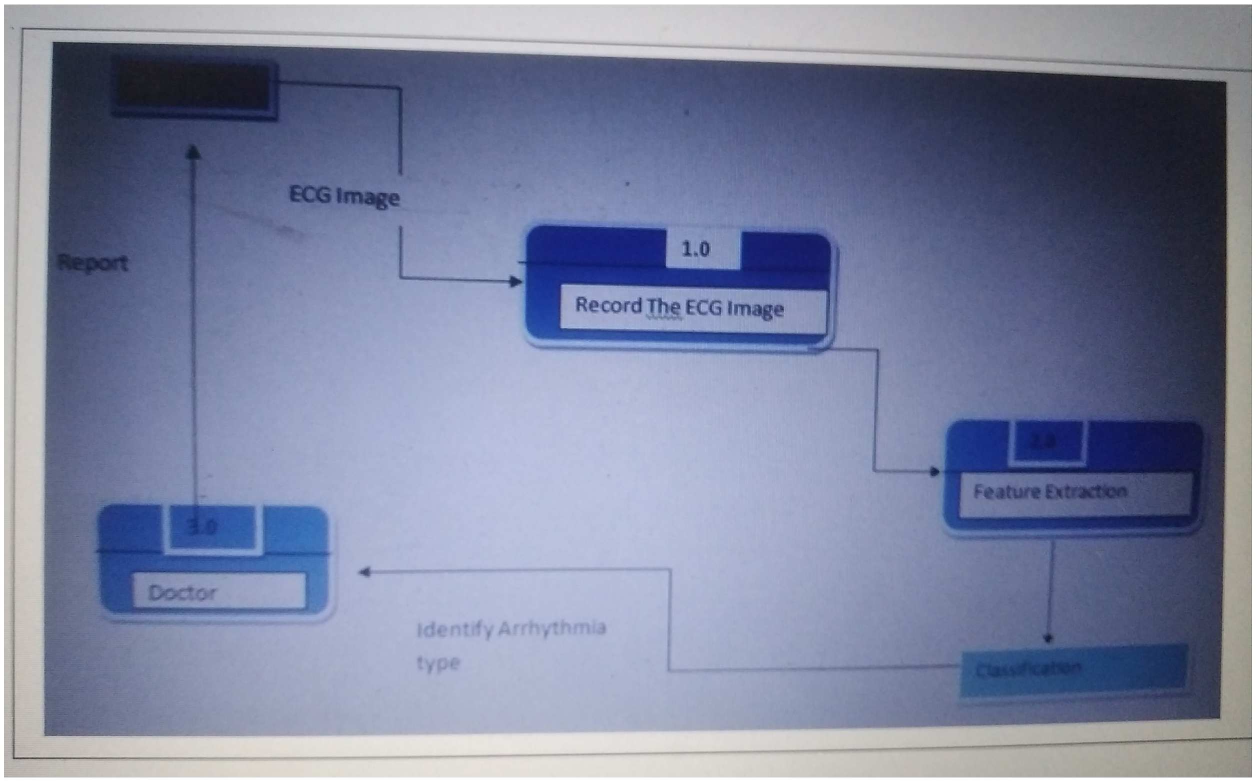
# 5. PROJECT DESIGN

## 5.1 Data Flow Diagrams :

A data flow diagram (DFD) is **a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement**. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method (SSADM). A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

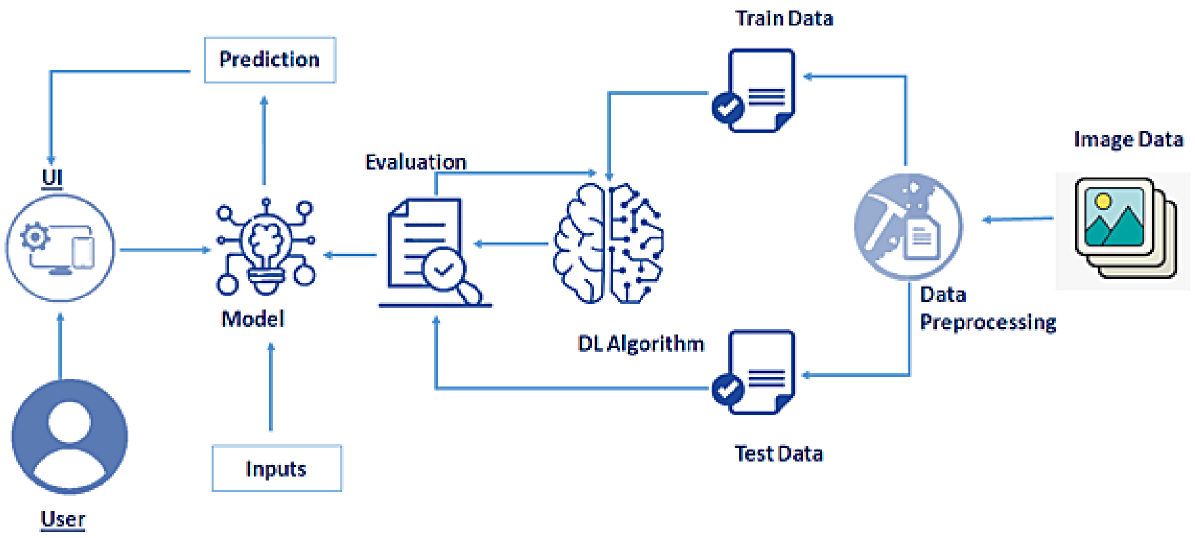
**Types :**

- 0 - Level DFD
- 1 - Level DFD
- 2 - Level DFD



## 5.2 Solution & Technical Architecture :

The input is Raw ECG which is given to signal denoising which filters out the image so that clear image is obtained. The output received from signal denoising is given to ECG Segmentation that segments ECG into five parts. The segmented ECG images are passed onto Bispectrum Analysis. From bispectrum analysis the features of the ECG are studied at 2-D CNN.and in features extraction. Through SVM-RBF classifier the required ECG for predicting Arrhythmia is found.



## 5.3 User Stories :

A user story is **an informal, general explanation of a software feature written from the perspective of the end user or customer**. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.

A user story or agile/ scrum user story is **a tool that's used in agile software development and product management to represent the smallest unit of work in the framework**. It provides an informal, natural language description of a feature of the software or product from the end-user perspective.

user stories for the product.

| **User Type** | **Functional Requirement(Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
|---|---|---|---|---|---|---|
| Doctor | Login | USN-1 | As a user I can logininto by email& password | | High | Sprint-1 |
| | | USN-2 | Upload the ECG Image | Access Image | High | Sprint-1 |
| ECG Techinician | Login | USN--3 | As a user I can logininto by email& password | To classify the image | High | Sprint-2 |

# 6. PROJECT PLANNING & SCHEDULING

## 6.1 Sprint Planning & Estimation :

- Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team.In scrum, the sprint is a set period of time where all the work is done. However, before you can leap into action you have to set up the sprint. You need to decide on how long the time box is going to be, the sprint goal, and where you're going to start. The sprint planning session kicks off the sprint by setting the agenda and focus. If done correctly, it also creates an environment where the team is motivated, challenged, and can be successful. Bad sprint plans can derail the team by setting unrealistic expectations.
- The What – The product owner describes the objective(or goal) of the sprint and what backlog items contribute to that goal. The scrum team decides what can be done in the coming sprint and what they will do during the sprint to make that happen.
- The How – The development team plans the work necessary to deliver the sprint goal. Ultimately, the resulting sprint plan is a negotiation between the development team and product owner based on value and effort.
- The Who – You cannot do sprint planning without the product owner or the development team. The product owner defines the goal based on the value that they seek. The development team needs to understand how they can or cannot deliver that goal. If either is missing from this event it makes planning the sprint almost impossible.
- The Inputs – A great starting point for the sprint plan is the product backlog as it provides a list of ‘stuff’ that could potentially be part of the current sprint. The team should also look at the existing work done in the increment and have a view to capacity.

- The Outputs – The most important outcome for the sprint planning meeting is that the team can describe the goal of the sprint and how it will start working toward that goal. This is made visible in the sprint backlog.
- Running a great sprint planning event requires a bit of discipline. The product owner must be prepared, combining the lessons from the previous sprint review, stakeholder feedback, and vision for the product, so they set the scene for the sprint. For transparency, the product backlog should be up-to-date and refined to provide clarity. Backlog refinement is an optional event in scrum, because some backlogs don't need it. However, for most teams, it's better to get the team together to review and refine the backlog prior to sprint planning.

## 6.2 Sprint Delivery Schedule :

Since sprints take place over a fixed period of time, it's critical to avoid wasting time during planning and development. And this is precisely where sprint scheduling enters the equation.

In case you're unfamiliar, a sprint schedule is a document that outlines sprint planning from end to end. It's one of the first steps in the agile sprint planning process—and something that requires adequate research, planning, and communication. In our project we schedule the sprint session into four parts.

Sprint - 1 : Collection Of Dataset

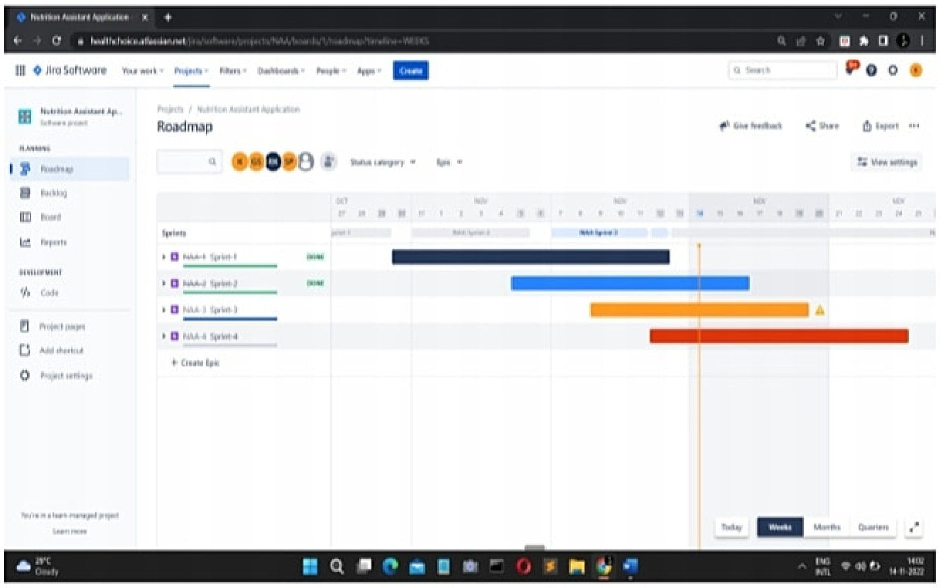
Sprint - 2 : Model Building

Sprint - 3 : Application Building

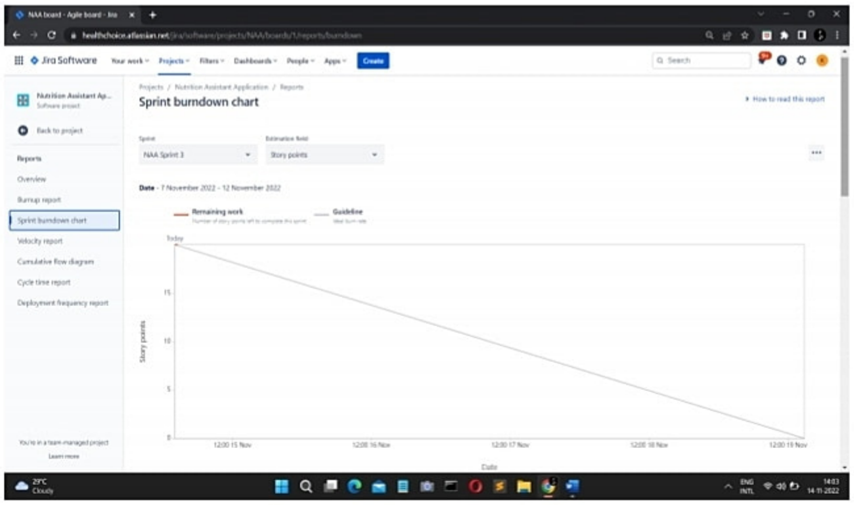
Sprint - 4 : User Acceptance Testing.

## 6.3 Reports From Jira :

JIRA Roadmap:



JIRA Burndown Chart:



# 7. CODING & SOLUTIONING

## 7.1 Feature 1 :

**Segmentation of image is the first feature .** The three primary forms of noise in the ECG signal are power line interference, baseline drift, and electromyographic noise. The noise from the original ECG signal must be removed to ensure that a denoised ECG signal is obtained for further processing. We combined wave let based thresholding and the reconstruction algorithm of wavelet decomposition to remove noise from the original ECG signal. The wavelet thresholding was performed using, Where, w,x,y represents the wavelet coefficients, ω,x,y represents the estimated wavelet coefficients after threshold, x represents the scale and y represents the shift, λ represents the threshold, and α is a parameter whose value can be set arbitrarily. The wavelet thresholding reduced the electromyographic noise and power line noise interference. Moreover, the reconstruction algorithm of wavelet decomposition was used to remove the baseline drift noise from the noisy ECG signal .

**Coding :**

```
    data = np.array(csv_data)
signals = []
count = 1
peaks =  biosppy.signals.ecg.christov_segmenter(signal=data, sampling_rate = 200)[0]
for i in (peaks[1:-1]):
    diff1 = abs(peaks[count - 1] - i)
    diff2 = abs(peaks[count + 1]- i)
    x = peaks[count - 1] + diff1//2
    y = peaks[count + 1] - diff2//2
    signal = data[x:y]
    signals.append(signal)
    count += 1
return signals
```

## 7.2 Feature 2 :

ECG Signal to image is the second feature. In this feature ecg signals is converted into images.

**Coding :**

```
for count, i in enumerate(array):
  fig = plt.figure(frameon=False)
  plt.plot(i)
  plt.xticks([]), plt.yticks([])
  for spine in plt.gca().spines.values():
     spine.set_visible(False)

  filename = directory + '/' + str(count)+'.png'
  fig.savefig(filename)
  im_gray = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
  im_gray = cv2.resize(im_gray, (128, 128), interpolation = cv2.INTER_LANCZOS4)
  cv2.imwrite(filename, im_gray)
```

# 8. TESTING

## 8.1 Test Cases :

A test case is a set of actions performed on a system to determine if it satisfies software requirements and functions correctly. It is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement.[1] Test cases underlie testing that is methodical rather than haphazard. A battery of test cases can be built to produce the desired coverage of the software being tested. Formally defined test cases allow the same tests to be run repeatedly against successive versions of the software, allowing for effective and consistent regression testing.

**Formal Test Case :** In order to fully test that all the requirements of an application are met, there must be at least two test cases for each requirement: one positive test and one negative test. If a requirement has sub-requirements, each sub-requirement must have at least two test cases. Keeping track of the link between the requirement and the test is frequently done using a traceability matrix. Written test cases should include a description of the functionality to be tested, and the preparation required to ensure that the test can be conducted. A formal written test case is characterized by a known input and by an expected output, which is worked out before the test is executed. The known input should test a precondition and the expected output should test a postcondition.

**Informal Test Case :** For applications or systems without formal requirements, test cases can be written based on the accepted normal operation of programs of a similar class. In some schools of testing, test cases are not written at all but the activities and results are reported after the tests have been run. In scenario testing, hypothetical stories are used to help the tester think through a complex problem or system. These scenarios are usually not written down in any detail.

## 8.2 User Acceptance Testing :

User acceptance testing (UAT), also called *application testing* or *end-user testing*, is a phase of software development in which the software is tested in the real world by its intended audience. UAT is often the last phase of the software testing process and is performed before the tested software is released to its intended market. The goal of UAT is to ensure software can handle real-world tasks and perform up to development specifications. In UAT, users are given the opportunity to interact with the software before its official release to see if any features have been overlooked or if it contains any bugs. UAT can be done in-house with volunteers, by paid test subjects using the software or by making the test version available for download as a free trial. The results from the early testers are forwarded to the developers, who make final changes before releasing the software commercially. UAT is effective for ensuring quality in terms of time and software cost, while also increasing transparency with users. UAT also enables developers to work with real cases and data, and if successful, the process can validate business requirements. This report showsthe number of resolved or closed bugs at each severity level, and how they were resolved

| Resolution | Severity 1 | Severity 2 | Severity 3 | Severity 4 | Subtotal |
|---|---|---|---|---|---|
| By Design | 10 | 4 | 2 | 3 | 20 |
| Duplicate | 1 | 0 | 3 | 0 | 4 |
| External | 2 | 3 | 0 | 1 | 6 |
| Fixed | 11 | 2 | 4 | 20 | 37 |
| Not Reproduced | 0 | 0 | 1 | 0 | 1 |
| Skipped | 0 | 0 | 1 | 1 | 2 |
| Won't Fix | 0 | 5 | 2 | 1 | 8 |
| Totals | 24 | 14 | 13 | 26 | 77 |

## Test Case Analysis

This report shows the number of test cases that have passed, failed,and untested

| Section | Total Cases | Not Tested | Fail | Pass |
|---|---|---|---|---|
| Print Engine | 7 | 0 | 0 | 7 |
| Client Application | 51 | 0 | 0 | 51 |
| Security | 2 | 0 | 0 | 2 |
| Outsource Shipping | 3 | 0 | 0 | 3 |
| Exception Reporting | 9 | 0 | 0 | 9 |
| Final Report Output | 4 | 0 | 0 | 4 |
| Version Control | 2 | 0 | 0 | 2 |

# 9. RESULTS

## 9.1 Performance Metrices :

All projects are feasible when given unlimited resources and infinite time. It's both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. The efforts and resource spent in developing the system will be waste if the end solution does not offer timely and satisfactory solutions to the user. Feasibility study is a test of system proposed regarding workability and effective use of resources. Thus when a new application is proposed, it normally goes through a feasibility study before it is approved for development. The feasibility of the project is analysed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out.

- Technical Feasibility
- Economical Feasibility
- Social Feasibility

**Technical Feasibility -** This study is carried out to check the technical feasibility that is the technical requirements of the system. Any system must not have a high demand on the available technical resources.

**Economical Feasibility -** This study is carried out to check the economic impact that the system will have on organization. The implementation of the system is very easy and simple when compared to other systems according to the cost and other factors.

**Social Feasibility -** The aspect of this study is to check the level the acceptance of the system. Those includes the process of training the user to use the system efficiently

# 10. ADVANTAGES & DISADVANTAGES

## 10.1 Advantages :

- Easy to maintain data.
- Less Time Consuming.
- Efficient Output.
- Reduce Labour work.
- Accuracy in result.
- Maintains Large Datasets.
- No paper work
- Reduce Human effort.

## 10.2 Disadvantages :

- Testing process takes time.
- Lab technicians only able to classify.

# 11. CONCLUSION

In this study, we proposed a 2-D CNN-based classification model for automatic classification of cardiac arrhythmias using ECG signals. An accurate taxonomy of ECG signals is extremely helpful in the prevention and diagnosis of CVDs. Deep CNN has proven useful in enhancing the accuracy of diagnosis algorithms in the fusion of medicine and modern machine learning technologies. The proposed CNN-based classification algorithm, using 2-D images, can classify eight kinds of arrhythmia, namely, NOR, VFW, PVC, VEB, RBB, LBB, PAB, and APC, and it achieved 97.91% average sensitivity, 99.61% specificity, 99.11% average accuracy, and 98.59% positive predictive value (precision). These results indicate that the prediction and classification of arrhythmia with 2-D ECG representation as spectrograms and the CNN model is a reliable operative technique in the diagnosis of CVDs. The proposed scheme can help experts diagnose CVDs by referring to the automated classification of ECG signals. The present research uses only a single-lead ECG signal. The effect of multiple lead ECG data to further improve experimental cases will be studied in future work

# 12. FUTURE SCOPE

In this we use only a single – lead ECG signal.The effect of multiple lead ECG data to further improve experimental cases will be studied in future work.

# 13. APPENDIX

## Source Code :

## 13.1 App.py

```
from __future__ import division, print_function
import json
# coding=utf-8
import sys
import os
import glob
import re
import numpy as np
import cv2
import pandas as pd
import numpy as np
import biosppy
import matplotlib.pyplot as plt
# Keras
from keras.applications.imagenet_utils import preprocess_input, decode_predictions
from keras.models import load_model
from keras.preprocessing import image

# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer

# Define a flask app
app = Flask(__name__)


# Model saved with Keras model.save()

# Load your trained model
model = load_model('path to the model')
```

```
model._make_predict_function()          # Necessary
print('Model loaded. Start serving...')
output = []
# You can also use pretrained model from Keras
# Check https://keras.io/applications/
#from keras.applications.resnet50 import ResNet50
#model = ResNet50(weights='imagenet')
#print('Model loaded. Check http://127.0.0.1:5000/')

def model_predict(uploaded_files, model):
    flag = 1

    for path in uploaded_files:
        #index1 = str(path).find('sig-2') + 6
        #index2 = -4
        #ts = int(str(path)[index1:index2])
        APC, NORMAL, LBB, PVC, PAB, RBB, VEB = [], [], [], [], [], [], []
        output.append(str(path))
        result = {"APC": APC, "Normal": NORMAL, "LBB": LBB, "PAB": PAB, "PVC": PVC,
"RBB": RBB, "VEB": VEB}


        indices = []

        kernel = np.ones((4,4),np.uint8)

        csv = pd.read_csv(path)
        csv_data = csv[' Sample Value']
        data = np.array(csv_data)
        signals = []
        count = 1
        peaks =  biosppy.signals.ecg.christov_segmenter(signal=data, sampling_rate = 200)[0]
        for i in (peaks[1:-1]):
           diff1 = abs(peaks[count - 1] - i)
           diff2 = abs(peaks[count + 1]- i)
           x = peaks[count - 1] + diff1//2
           y = peaks[count + 1] - diff2//2
        signal = data[x:y]
        signals.append(signal)
        count += 1
        indices.append((x,y))


for count, i in enumerate(signals):
    fig = plt.figure(frameon=False)
    plt.plot(i)
    plt.xticks([]), plt.yticks([])
    for spine in plt.gca().spines.values():
        spine.set_visible(False)

    filename = 'fig' + '.png'
    fig.savefig(filename)
    im_gray = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
    im_gray = cv2.erode(im_gray,kernel,iterations = 1)
    im_gray = cv2.resize(im_gray, (128, 128), interpolation = cv2.INTER_LANCZOS4)
    cv2.imwrite(filename, im_gray)
    im_gray = cv2.imread(filename)
    pred = model.predict(im_gray.reshape((1, 128, 128, 3)))
    pred_class = pred.argmax(axis=-1)
    if pred_class == 0:
        APC.append(indices[count])
    elif pred_class == 1:
        NORMAL.append(indices[count])
    elif pred_class == 2:
        LBB.append(indices[count])
    elif pred_class == 3:
        PAB.append(indices[count])
    elif pred_class == 4:
        PVC.append(indices[count])
    elif pred_class == 5:
        RBB.append(indices[count])
    elif pred_class == 6:
        VEB.append(indices[count])
        result = sorted(result.items(), key = lambda y: len(y[1]))[::-1]
        output.append(result)
        data = {}
        data['filename'+ str(flag)] = str(path)
        data['result'+str(flag)] = str(result)

        json_filename = 'data.txt'
        with open(json_filename, 'a+') as outfile:
            json.dump(data, outfile)
        flag+=1




    with open(json_filename, 'r') as file:
        filedata = file.read()
    filedata = filedata.replace('}{', ',')
    with open(json_filename, 'w') as file:
        file.write(filedata)
    os.remove('fig.png')
    return output






@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        uploaded_files = []

        # Save the file to ./uploads
        print(uploaded_files)
        for f in request.files.getlist('file'):

            basepath = os.path.dirname(__file__)
            file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
            print(file_path)
            if file_path[-4:] == '.csv':
                uploaded_files.append(file_path)
                f.save(file_path)
        print(uploaded_files)
        # Make prediction
        pred = model_predict(uploaded_files, model)


        # Process your result for human
                    # Simple argmax
        #pred_class = decode_predictions(pred, top=1)   # ImageNet Decode
        #result = str(pred_class[0][0][1])               # Convert to string
        result = str(pred)


        return result
    return None


if __name__ == '__main__':
    # app.run(port=5002, debug=True)

    # Serve the app with gevent
    http_server = WSGIServer(('', 5000), app)
    http_server.serve_forever()
```

## 13.2 info.html

```
<html>
<head>
<meta name="viewport" content="width=device-width, minimum-scale=0.1">
<title>Predicted type</title>
</head>
<body style="margin:0px; background=#0e0e0e; height=100%>
</body>
</html>
```

**Github Link :** IBM-EPBL/IBM-Project-42627-1660671148: Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation (github.com)

**Demo Video Link :**

Project Demo.mp4 - Google Drive