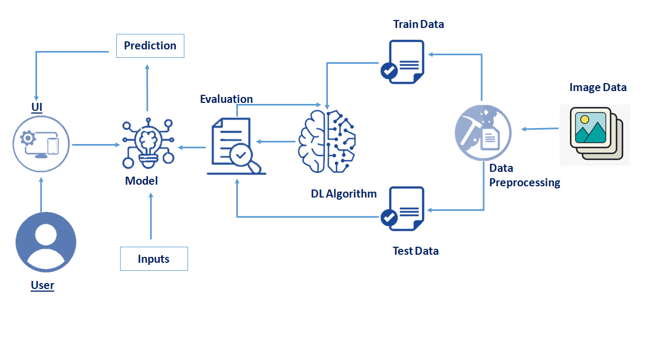
**Classification Of Arrhythmia By Using Deep Learning With 2-D ECG Spectral Image Representation**

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.

**Technical Architecture:**



**Project Objectives**

**By the end of this project you will:**

* Know fundamental concepts and techniques of the Artificial Neural Network and Convolution Neural Networks
* Gain a broad understanding of image data.
* Work with Sequential type of modeling
* Work with Keras capabilities
* Work with image processing techniques
* know how to build a web application using the Flask framework.

**Project Flow**

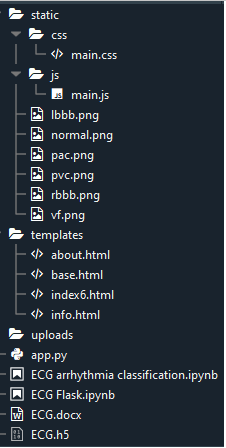
* User interacts with User interface to upload image
* Uploaded image is analyzed by the model which is integrated
* Once model analyses the uploaded image, the prediction is showcased on the UI

To accomplish this, we have to complete all the activities and tasks listed below

* Data Collection.
  + Collect the dataset or Create the dataset
* Data Preprocessing.
  + Import the ImageDataGenerator library
  + Configure ImageDataGenerator class
  + Apply ImageDataGenerator functionality to Trainset and Testset
* Model Building
  + Import the model building Libraries
  + Initializing the model
  + Adding Input Layer
  + Adding Hidden Layer
  + Adding Output Layer
  + Configure the Learning Process
  + Training and testing the model
  + Optimize the Model
  + Save the Model
* Application Building
  + Create an HTML file
  + Build Python Code

**Project Structure**

Create a Project folder which contains files as shown below



* We are building a Flask Application that needs  HTML pages stored in the templates folder and a python script app.py for serverside scripting
* 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, those images are saved in the uploads folder.

**Prerequisites**

**To complete this project you should have the following software  and packages**

**Anaconda Navigator :**

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform,  package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupiter notebook and spyder

To install Anaconda navigator and to know how to use Jupyter Notebook a Spyder using Anaconda watch the video

To build Deep learning models you must require the following packages

**Tensor flow:**TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers can easily build and deploy ML powered applications.

**Keras :** Keras leverages various optimization techniques to make high level neural network API easier and more performant. It supports the following features:

* Consistent, simple and extensible API.
* Minimal structure - easy to achieve the result without any frills.
* It supports multiple platforms and backends.
* It is user friendly framework which runs on both CPU and GPU.
* Highly scalability of computation.

**Flask:** Web frame work used for building  Web applications

Watch the below video to Install the necessary Packages

### Prior Knowledge

**Supervised and unsupervised learning:**

Watch the below video to know about the types of machine learning

Dataset Collection

Artificial Intelligence is a data hunger technology, it depends heavily on data, without data, it is impossible for a machine to learn. It is the most crucial aspect that makes algorithm training possible. In Convolutional Neural Networks, as it deals with images, we need training and testing data set. It is the actual data set used to train the model for performing various actions. In this activity lets focus of gathering the dataset

### Download The Dataset

You can collect datasets from different open sources like kaggle.com, data.gov, UCI machine learning repository, etc.

 The dataset used for this project is in this [Link](https://drive.google.com/file/d/16SUrk6lMaakmVf4axGNDub3joHl-XdBT/view?usp=sharing)  Please refer to the link to download the data set 

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

https://ssl.gstatic.com/images/branding/product/1x/drive_2020q4_32dp.png

<https://drive.google.com/file/d/16SUrk6lMaakmVf4axGNDub3joHl-XdBT/view?usp=sharing>