**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**DeCode EKG**

**Introduction**

The standard 12-lead electrocardiogram (ECG/ EKG) is one of the most commonly used medical studies in the assessment of cardiovascular disease. It is the most important test for interpretation of the cardiac rhythm, detection of myocardial ischemia and infarction1, conduction system abnormalities, preexcitation, long QT syndromes, atrial abnormalities, ventricular hypertrophy, pericarditis, and other conditions. The use of standardized terminology is recommended to allow application of ECG diagnoses across many different hospital systems and countries.

**Overview of the idea**

There are several ongoing efforts to enable new class of EKG devices where captured EKG data is analysed without requiring the need to print the EKG on paper. The deployment of such systems is limited and may not be available to larger set of users. More so in India, there is a large existing deployment of cost effective & conventional EKG devices in cities & villages. These conventional systems print the EKG report and the report can be later be analysed only by a cardiologist.

In a country like India, we envision a genuine need for larger population to leverage existing medical deployments, need to augment broader availability of smartphones to deliver medical intelligence as no infrastructural cost.

**Value proposition**

The use of AR in medicine has grown significantly. The idea is to develop an AR application to assist clinical decision making. The existing EKG devices deployed in cities and villages can be reused as is but the interpretation of reports can be extended beyond cardiologist with intelligence on mobile phone.

The product as a whole will be an augmented reality application to read paper forms of EKGs via camera and display -

• Heart Rate

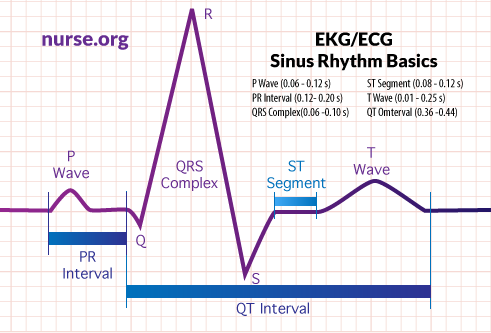
• waves amplitudes P, Q, R, S, T;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

• intervals RR, PQ, QRS, QT, QTс;

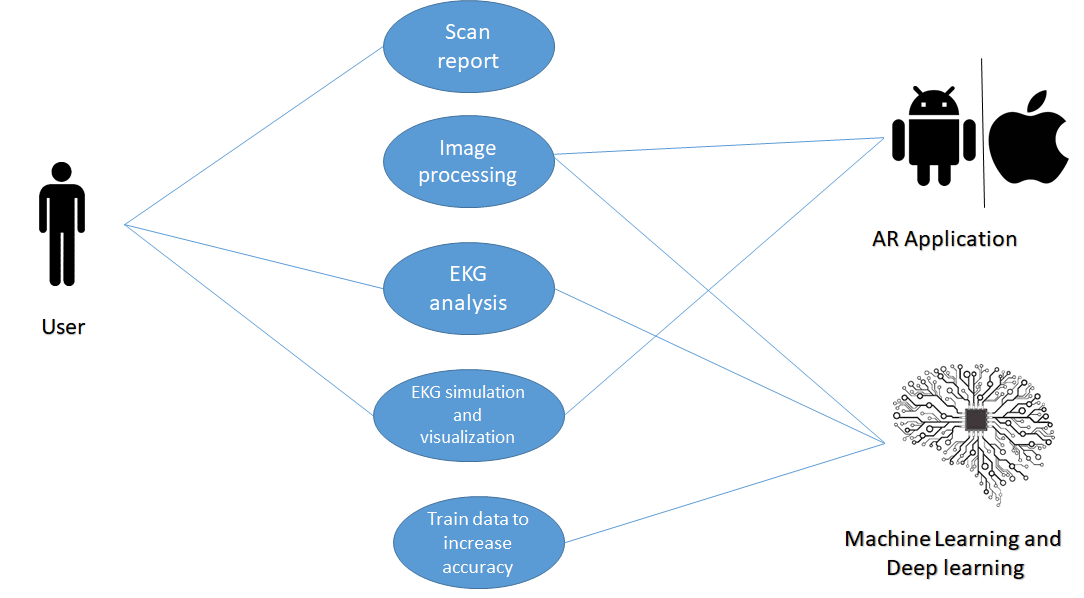
• the position of the segment ST.

And draw suitable conclusions with live simulation (via AR). We tend to develop an application in future that scans the EKG report and the AR application will provide insights just like the recent EKG monitoring device used in hospital.

Thus the application will be a great use to not only to common people but also doctors to infer, simulate and visualize the plots/ nature of waves and provide insights and draw inferences.

**Target market**

TAM (Total Addressable Market) : Doctors and Everyone with a smartphone

**Use case diagram**

**Technology stack**

1. **Application Development :**

Frontend

* Android Studio
* Flutter / Dart Programming
* Android/ IOS

Backend

* Firebase
* Google Console

1. **Machine Learning:**

Google Colabs

* OpenCV (Morphological and image processing tools)
* Convolutional Neural Network (CNN) YOLO detection fully connected layers

1. **Augmented Reality** (Yet to be implemented)

* Amazon Sumerian
* Unity

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Current work done**

We came up with the idea during the hackathon. Our first aim was to detect the R-R peaks and then measure the interval.

This is how we progressed with the work:

**I) Preparation of data set:**

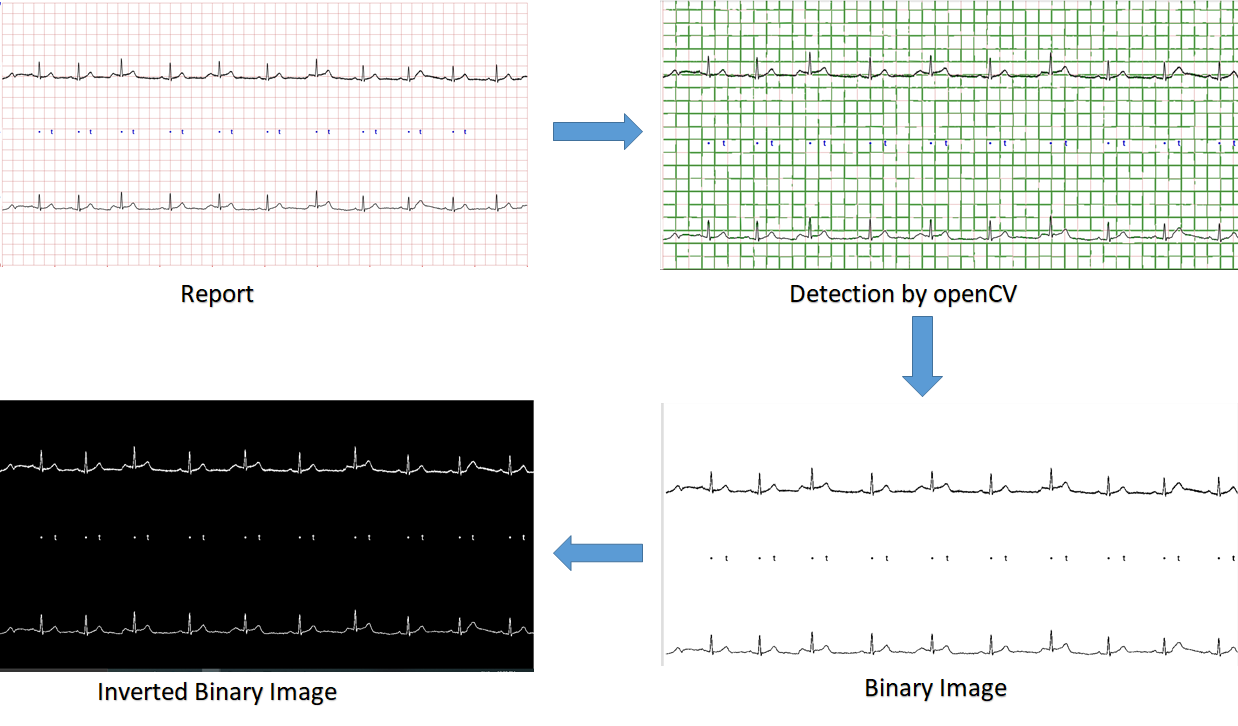
The site ‘<https://physionet.org/lightwave/>’ provides a 20 second recording of each person. We took the screenshots of 10 second recording of each patient followed by addition of annotations of the R peaks. Each image was cropped into the same pixel size (to form the training dataset) and processed individually through the image processing model to record the R and T peaks and stored in a csv file.

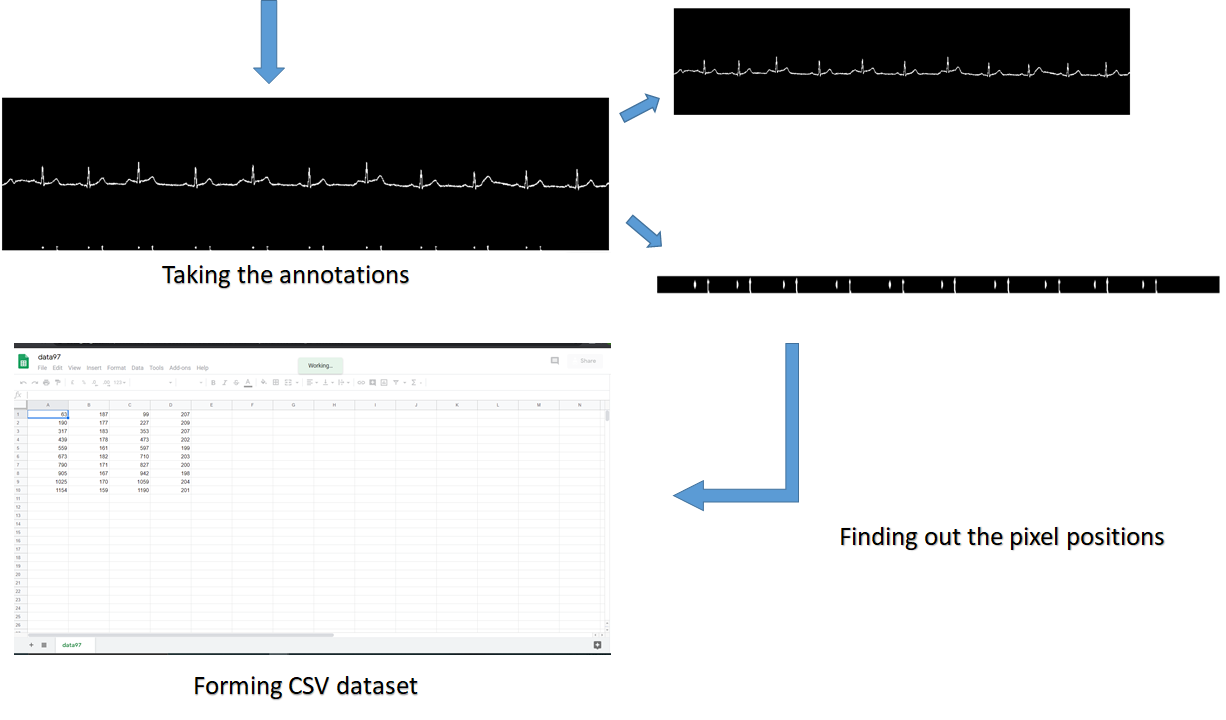


**II) Image processing:**

For simplicity we are limiting the scope of model to locating R and T peaks (to be precise we are fine tuning are model for R peaks particularly. Then we wil move forward to other parameters.)

Here is a detailed description of the above process Annotating the images and locating the R and T points using image processing and storing the locations of R and T peaks into csv file. We determine the R and T peak points using ‘.’ and ‘t’ annotations in the image which represent R and T peakpoints respectively. We are concerned with only predicting The time intervals at which these peaks occur rather than their peak values. The above images were converted to binary images.We then cropped each signal image to 200X30 pixels which we will be using for training purposes. We will be using sliding window approach to make predictions in the final input image.



****

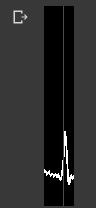
Following is a sample cropped image(200X30 pixels) with an R peak. These type of images will be used for training. :



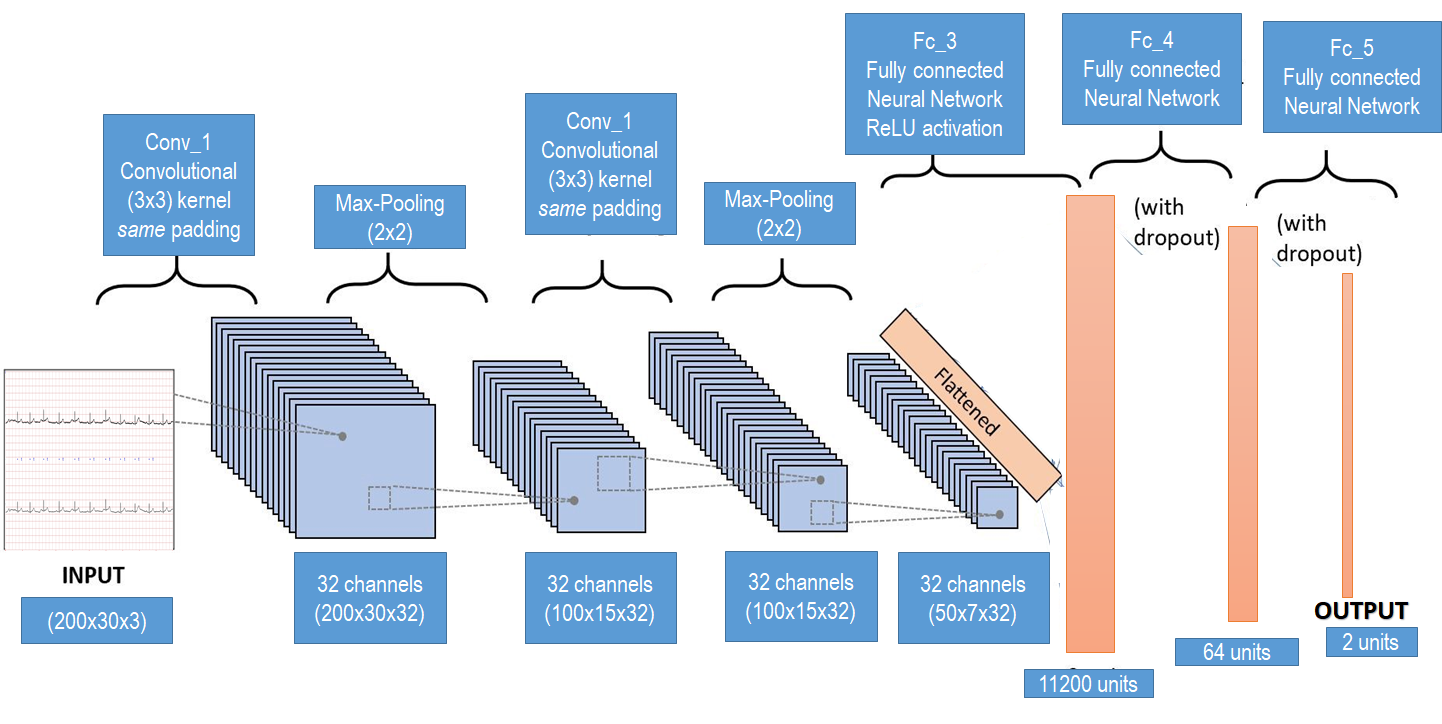
**III) Deep Learning**

We have provided the final Training Dataset with the file names input.npy and target.csv The input.npy contains numpy array of the training images (each image is 200x30 pixels) and target.csv contains the ground truth values. Each row of target.csv is of the form [is there R peak in the given image, is there T peak in the given image, x\_location(time) of R peak, x\_location(time) of T peak] i.e. the file contains 4 columns. The first two elements of each row of target.csv (or the values of first two columns ) can either be 0 or 1 and the last two elements’ values (or the values of last two columns) range from 0 to 1 (pixels 0 to 30 have been normalised from 0 to 1). Hence we use categorical cross entropy loss function for for first two elements of the above array, and mean squared error loss function for the last two elements.

We have used 2 stages of Conv2D (kernel(3,3)) followed by Max-Pooling(pool-size=2,2) layer. These two stages are followed by 2 Dense fully connected layers for the first two elements and 2 different Dense fully connected layers for predicting the last two elements. The training accuracy achieved is around 73% (for 20 epochs) for the first two elements and validation accuracy is around 60%. The above mentioned accuracy is for model1(model1.h5) which is used for prediction of the first two columns of target.csv. For predicting the last two columns we use model2(model2.h5). The training accuracy of model2 is around 47% and validation accuracy is around 52%. *(you may refer to Training.ipynb or Pred\_Model.ipynb for understanding model1 and model2).* Due to time constraints we did not get sufficient time to fine tune parameters like Neural Network Architecture, number of Layers, activation functions, etc. Following is a sample of correct prediction by our model: (Green line is our prediction of locating R peak.)

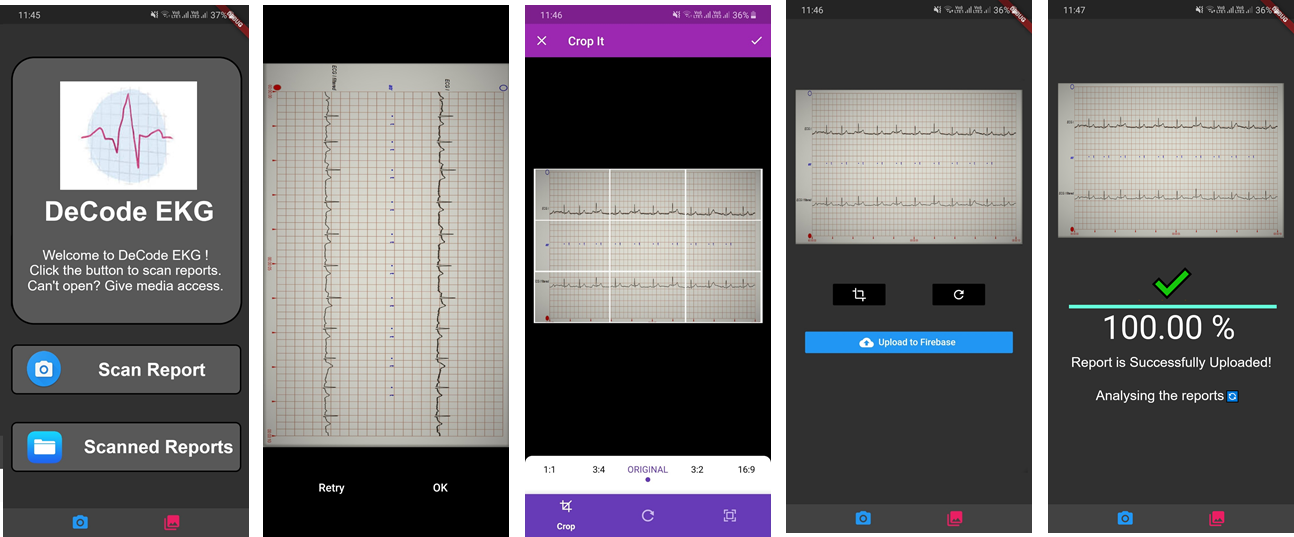


There are some incorrect predictions too which can be minimized by playing with the architecture more and greater accuracy can be achieved good enough to deploy the model on the final platform. We can add BatchNormalization layer after relu instead of sigmoid layer for better performance. Also the current model is trained only on 443 images which are like the one shown above. You may refer to the following architecture diagram for reference to our architecture. We can also add BatchNormalization layer after relu instead of sigmoid layer.

****

**IV) Application Development:**

A flutter application developed which can take the image and reshape/ resize it. (crop, scale and rotate)**.**  The image can then be uploaded on the firebase.



*The apk of the app has been uploaded on the drive.*

**Extensions to current work**

Further steps:

* **Machine Learning model:**

1. Training the model over more data and increase its accuracy
2. Detect the pq and st complex
3. Analyze these complexes
4. Entire pqrst analysis

* **For the application:**

1. Link firebase to real-time database to fetch the image to the model through API *(due to time constraints we couldn't add that for now)*
2. Work on the UI of the application
3. Overlay the RR peaks on live camera

* **Integration of Augmented Reality:**

Making use of Amazon Sumerian and Unity to create the AR application.

It would provide live simulations like the ECG device and some extensions to it to visualize it.

*(--Thought still in process-- )*

**Extension to current idea**

The idea can be extended to provide analysis of individual’s cardiac health (eg: risk involved).Also by taking real time pulses and analyzing it, many things can be determined like the mood of the person etc.

**Conclusion**

EKG inferencing with Artificial Intelligence continues to be an area of deep interest and leading to development of new smart hardware devices, there are fundamental economic needs to enable parallel solutions for cost sensitive geographies. Our proposal is a small baby step towards such a larger vision, starting with augmented reality based EKG inferencing that leverages existing devices and paper reports and enables analytics in the hands of every person. While the work is primitive at this stage, we hope it brings motivation in the community to build several other solutions, beyond EKG, that can be quickly and easily adopted in countries like India at no cost to end users.

**References**

I] <https://www.nature.com/articles/s41591-018-0268-3?WT.feed_name=subjects_machine-learning>

II] <https://physionet.org/about/database/>

III] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6085199/>

IV] <https://geekymedics.com/how-to-read-an-ecg/>

V] <https://nurse.org/articles/how-to-read-an-ECG-or-EKG-electrocardiogram/>

VI] <https://arxiv.org/abs/1804.02767>