CS523 Project
ECG Arrhythmia Classification with DNN

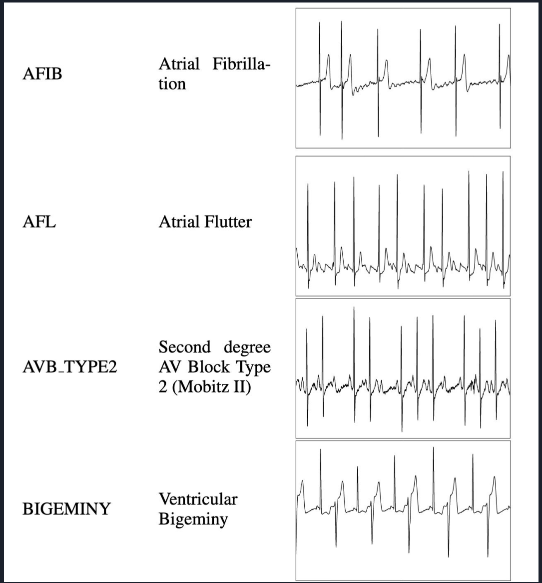
Anthony Sayegh and Alvaro Carrascosa Penabad

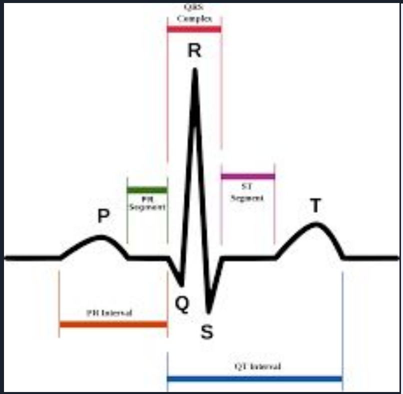
Motivation

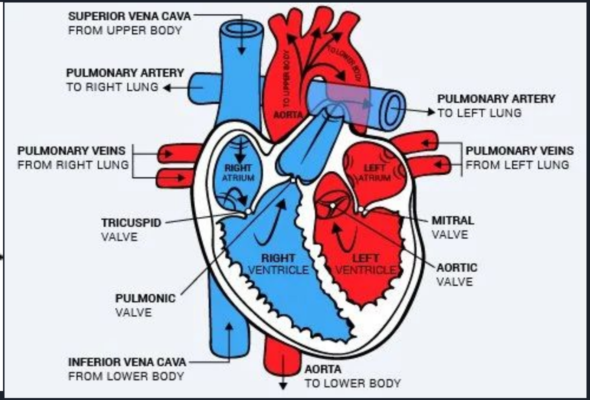
- Detection and diagnosis of irregular heart rhythm can save people lives.
- Traditional Methods of analyzing ECGs have been around since the 1980s. Many of them involve long complex DSP techniques that struggle to extract the details from ECGs to provide accurate predictions.
- Big data has created more labeled medical data that can be used to develop classification algorithms. Usually the amount of data on these conditions is scarce.
- These algorithms can help cardiologist in diagnosing conditions, and can make access to a cardiologist easier.

Background on ECGs and Arrhythmias

- ECG has a distinct repeating signal structure where each interval corresponds to an action performed by the heart.
- Arrhythmias are abnormal and irregular activity of the heart
  - Not all arrhythmias are dangerous, some are benign.
  - Some have unpleasant side effects and increase the risk of disease







- https://arxiv.org/pdf/1707.01836.pdf
- https://en.wikipedia.org/wiki/Sinus_rhythm
- https://www.medicinenet.com/heart_how_the_heart_works/article.htm

Goals

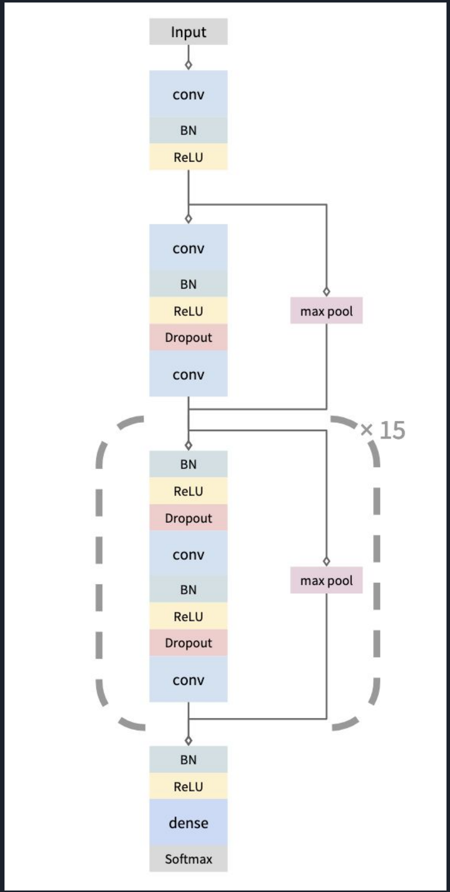
- Reproduce a basic model that can classify arrhythmia from a raw ECG and more complex model.
  - We tried 2 versions, both end-to-end DNN
- Understand tradeoffs between models in terms of performance, complexity, and capability.

Main Reference Work

We referenced our work from a few sources:

1. Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks
    a. Used data set collected by Irythim (~64,000 records from ~30,000 Patients) but have access to 10x more patients. (Need to request, and can't make public)
    b. 34 Layer Residual Neural Network
    c. 14 Classes
    d. Comparable performance to individual Cardiologist
2. Classification of Arrhythmia by Using Deep Learning with 2D ECG Spectral Image Representation
    a. Compute the spectrogram of each ECG and feed into CNN
    b. Similar to automatic speech recognition(ASR)/wake word detection pipeline.
    c. 7 Classes
3. Physionet/CINC Challenge 2020 (Kaggle)
    a. This entry used a RESNET model to get great results
    b. Tried a ResNet, Encoder/VGGNet, and CNN
    c. Large dataset
    d. 27 Classes

We found over 100 different algorithms for ECG detection! (All used one or more of the following: DSP, ML, DNN)

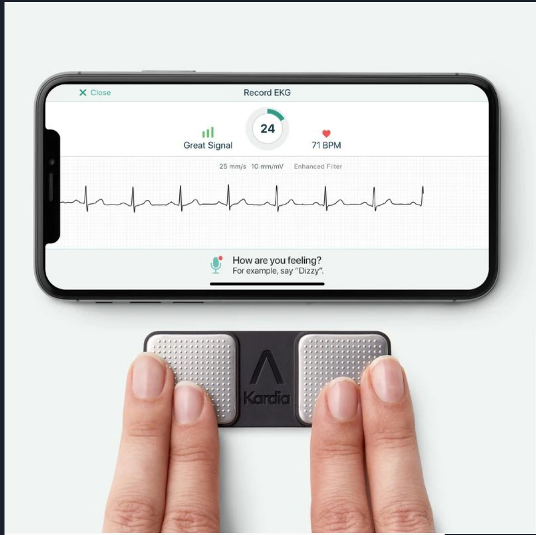


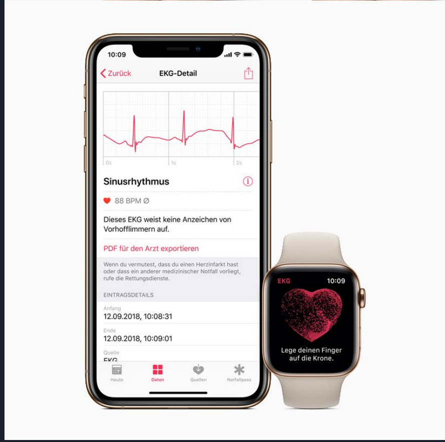
- https://arxiv.org/pdf/1707.01836.pdf

Design Considerations

- Scale of Algorithms
  - Detector vs. Diagnoser
- Product integration
  - Small networks for wearable devices/phones
  - Many people might want to keep medical data out of the cloud.
- Depth of network corresponding architecture
  - #number of parameters
  - Accuracy needed and latency(real-time applications)

- https://www.apple.com/newsroom/2019/03/ecg-app-and-irregular-rhythm-notification-on-apple-watch-available-today-across-europe-and-hong-kong/
- https://store.alivecor.com/products/kardiamobile?bvstate=pg:2/ct:r



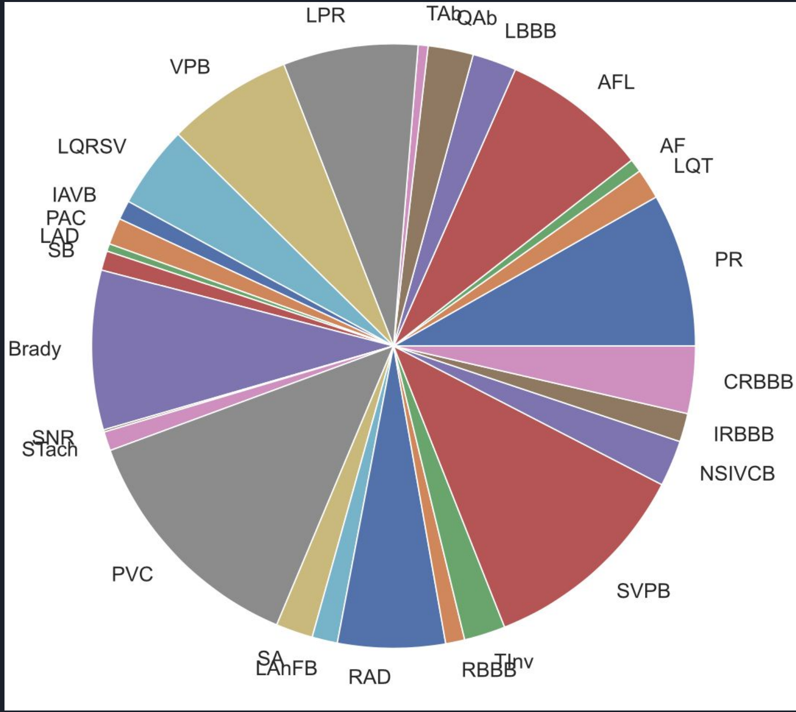


Architecture(Detector)

- Simple CNN (maxpooling layers)
- Input Dimensions = 187 (Driven by CSV file) at 125sps → ~ 1.5s of ECG
- Loss function = Cross Entropy
- MIT-BIH database (5 classes, > 100,000 labeled data points)
- Data was mostly normal sinus rhythms
- Lower F1 score and confusion matrix was

Architecture(Diagnoser)

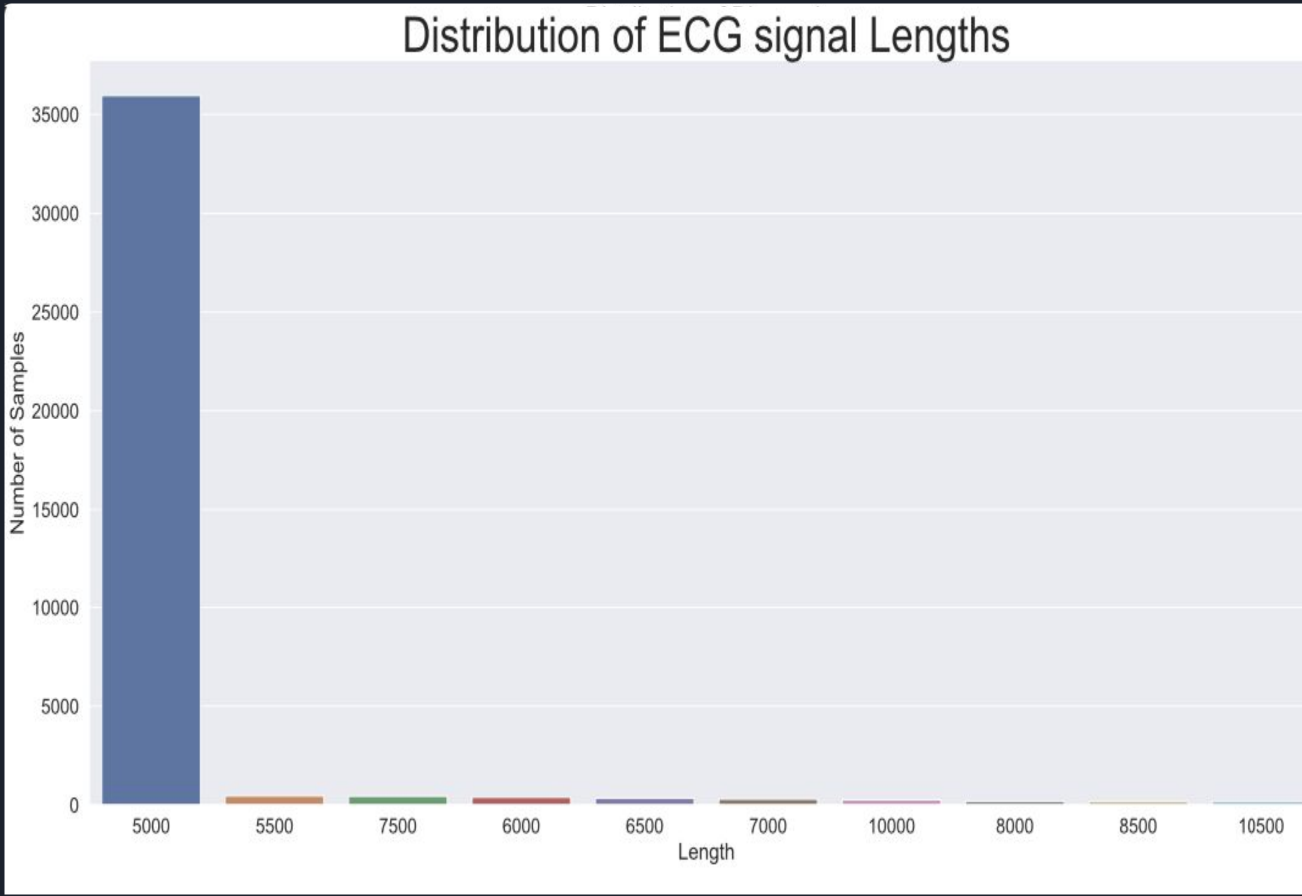
- Residual Neural Network ResNet
  - Depth
  - Input
  - Output
  - Loss function
  - Classify ECG waveforms to 27 Classes
  - K-Fold Split
  - Batch Generator
  - Imbalanced Data
  - Learning Rate Reduction
  - Early Stopping



- https://arxiv.org/pdf/1707.01836.pdf

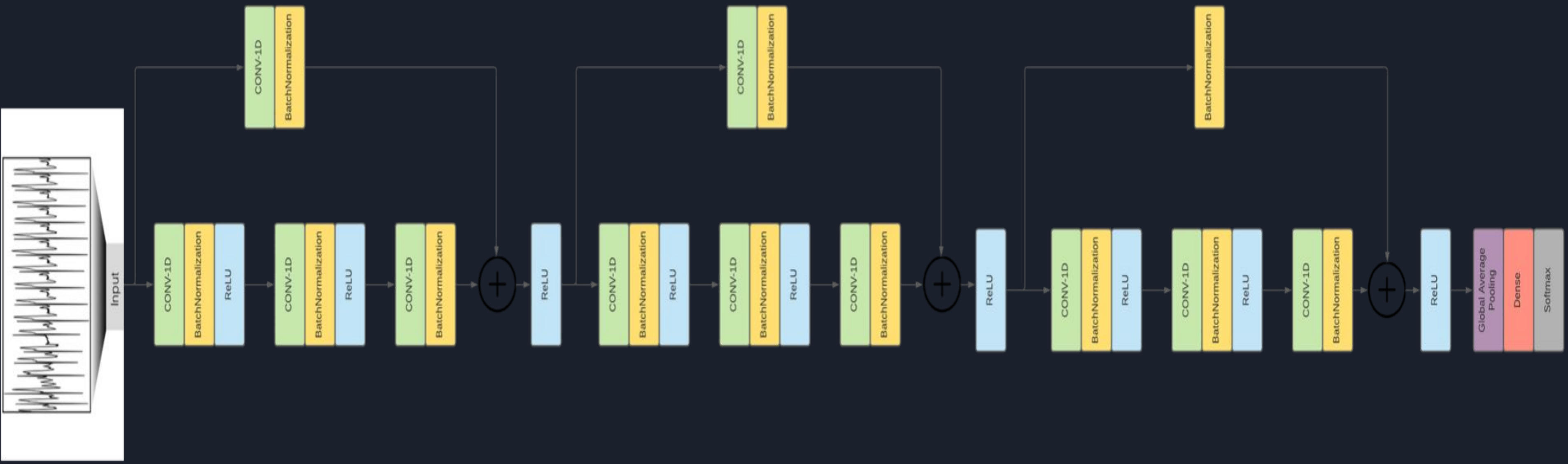
Data

- ~10GB
- Sources
- Format
- Length Distribution
- Class Distribution



- https://www.kaggle.com/bjoernjostein/physionet-challenge-2020/

Model



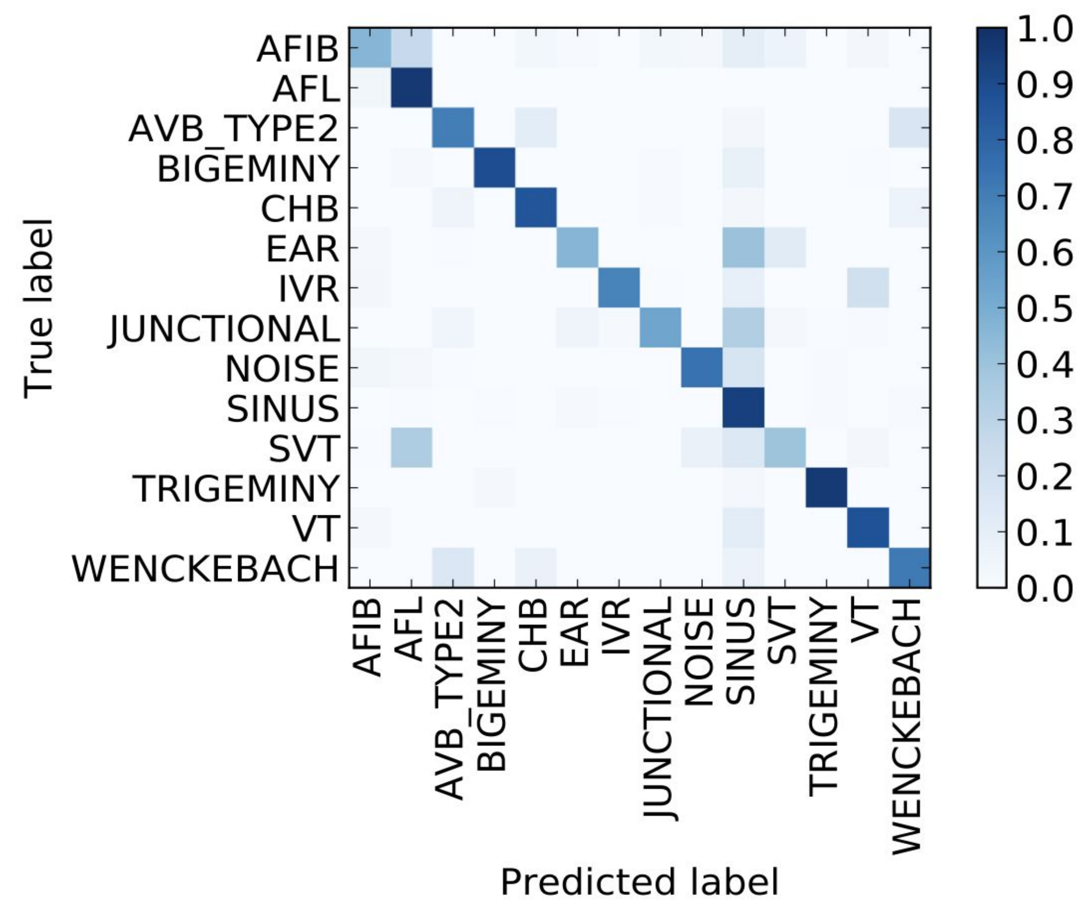
Input
CONV-1D
BatchNormalization
ReLU
Global Average Pooling
Dense
Softmax

Experimental Setup

- 8-Layer CNN
  - Code was built using Tensorflow and run on Google Colab with GPU support for fast training time
  - 12 Seconds per epoch and training batch size of 100
- 11-Layer ResNet
  - Code was built using Tensorflow and run on our personal machine with GPU support, taking around 10 minutes per epoch and a batch size of 30, for a total of 100 epochs.

Results

- Dataset composition made a huge difference for the models we looked at.



- https://www.kaggle.com/bjoernjostein/physionet-challenge-2020/
- https://arxiv.org/pdf/1707.01836.pdf

Demo

Github Link (WIP)

https://github.com/varocarras/ECG-523

References

- https://arxiv.org/pdf/1707.01836.pdf
- https://en.wikipedia.org/wiki/Sinus_rhythm
- https://github.com/hsd1503/DL-ECG-Review
- https://physionet.org/content/challenge-2017/1.0.0/
- https://store.alivecor.com/products/kardiamobile?bvstate=pg:2/ct:r
- https://www.irhythmtech.com/patients/how-it-works
- https://www.apple.com/newsroom/2019/03/ecg-app-and-irregular-rhythm-notification-on-apple-watch-available-today-across-europe-and-hong-kong/
- https://www.medicinenet.com/heart_how_the_heart_works/article.htm
- https://www.kaggle.com/bjoernjostein/physionet-challenge-2020/

Questions?