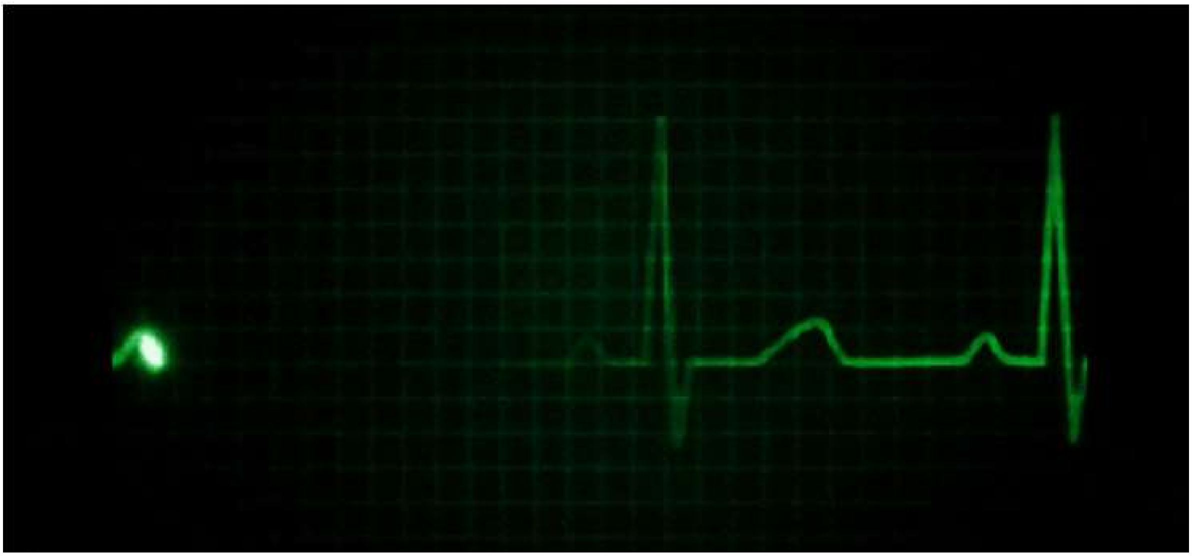
ECG classification
with CNN models



046211 - Deep Learning - Project Report

Name	ID	Campus E-Mail
Gilad Altshuler	315071621	gilada@campus.technion.ac.il
Sagie Badoach	211771746	sagie.ba@campus.technion.ac.il

Project Name:

ECG heartbeat classification with Convolution Neural Networks.

Abstract:

In this project, we analyze the method based on deep convolutional neural networks for the classification of heartbeats which can accurately classify five different arrhythmias in accordance with the AAMI EC57 standard and a method for transferring the knowledge acquired on this task to the myocardial infarction (MI) binary classification task.

Introduction:

Electrocardiogram (ECG) can be reliably used as a measure to monitor the functionality of the cardiovascular system.

ECG heartbeat classification is a hot-topic task to classify a time vector of ECG signal, to one heart condition out of 5 possible based on AAMI EC57 categories. Each category implies on some heart condition. The process of generating the signals and labeling described as follows:

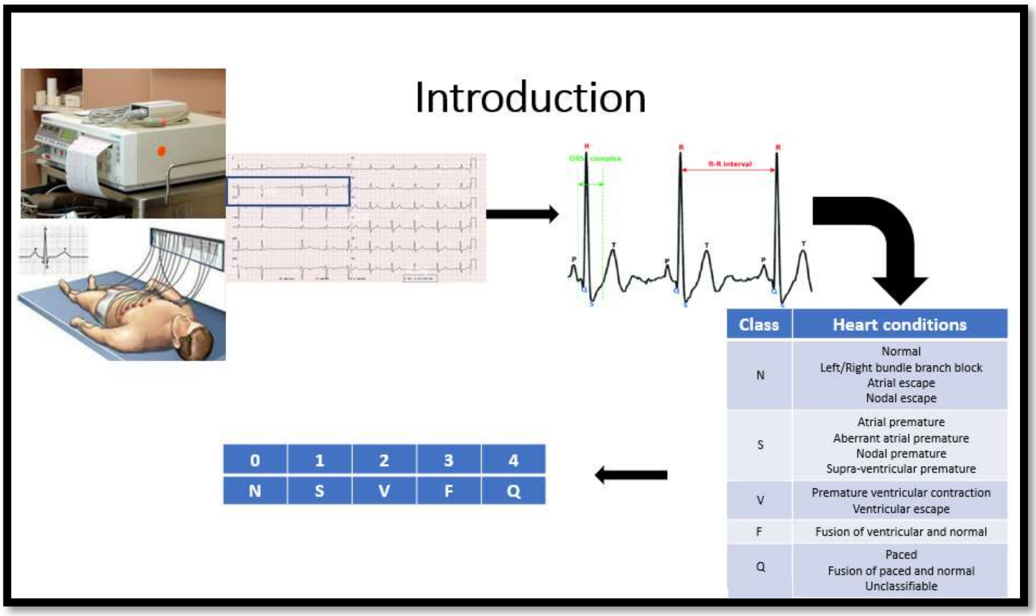


Figure 1: The process of generating the signals and labeling.

Our main goals throughout this project, were 3:

- Design simple yet accurate model to the classification task.
- Try to transfer the learnt model for Arrhythmia task for MI task.
- Comprehend the CNN layers once the model trained and their meanings.

Our motivation for these goals is twice-fold:

1) There is much influence and meaning in the topic, because it is benefits the accuracy of predicting subject with his / her condition.
2) We assume that the data within the ECG signal lies in lower dimension than the continuous time vector. We assume that data mainly encoded by intervals between some peaks in the ECG signal. So, if this true, simple, transferable and meaningful model can be trained well.

Until 2018, no accurate or significant results have shown to this task. In 2018, Kachuee et Al, proposed a CNN model method, including data preprocessing steps, which take no assumptions on the medical knowledge (Kachuee et Al, 2018). Their initial model comprised of 13 layers of CNN, which contains 5 residual blocks.

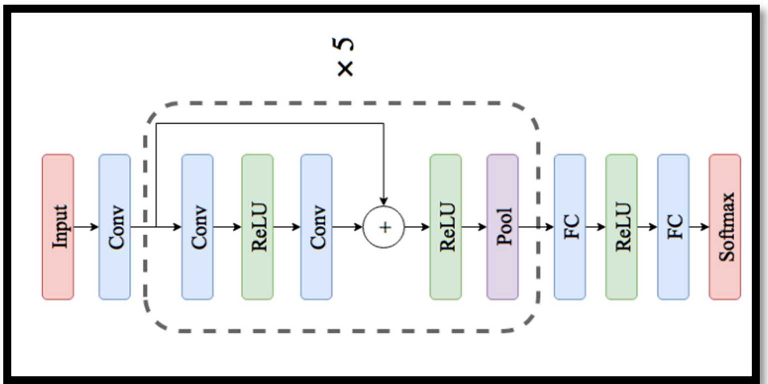


Figure 2: The initial CNN model for the task (Kachuee et Al, 2018).

The accuracy was significantly higher than the previous works, and they achieved 93.4% accuracy. Since then, much work have done, and many showed CNN model (which are simple enough) that reaches close to 100% accuracy (for example - [ECG-CNN-98.6% accuracy | Kaggle]). Many of sourced listed in acknowledgement used the data augmentation, including slicing, zero padding ignoring, and shifting, to massage the problem of the imbalanced data. We will address this issue latter. Also, Kachuee et Al and others used transferring method including auto-tuning, and freezing, and got good results on datasets. The main trend for all Kaggle, github, etc, codes was as follows – as complex the model gets, the accuracy didn't changed significantly, and even simple models got good accuracy results.

Kachuee et Al, conducted TSNE mapping on their last convolutional matrix and found good separability on the Arrhythmia dataset classes.

method:

Throughout the whole analysis method, we used CNN to train and classify the inputs.

CNN - Convolution Neural Network, is a network that based on the convolution idea, and comprised of convolution layer. Each CN layer, includes number of filters, which striding on the previous neurons / input, and therefore calculating a function of adjacent cells (which the number of cells can be controlled). Assumingly, By doing this, each layer, aided with batch normalization and activation functions, is able to get some space-time resolution, so each deeper layer is tend to be more general, and with lower resolution, but higher in perception.

After the CN layers, usually comes fully connected layer; we did so in our model. For classification task like our case, the last layer contain number of output classes of entities, which then a loss can be calculated on them, by performing a softmax function on them.

We chose the cross-entropy loss with class size weightening due to some imbalances (see in dataset section), which is trivially selected with the model setting (multiclass, softmax, cnn, etc).

First we tried to train same model as Kachuee et Al (see figure 2) which turned out to be too complex for the problem.

We suggested the model architecture as follows (zoom in for better resolution):

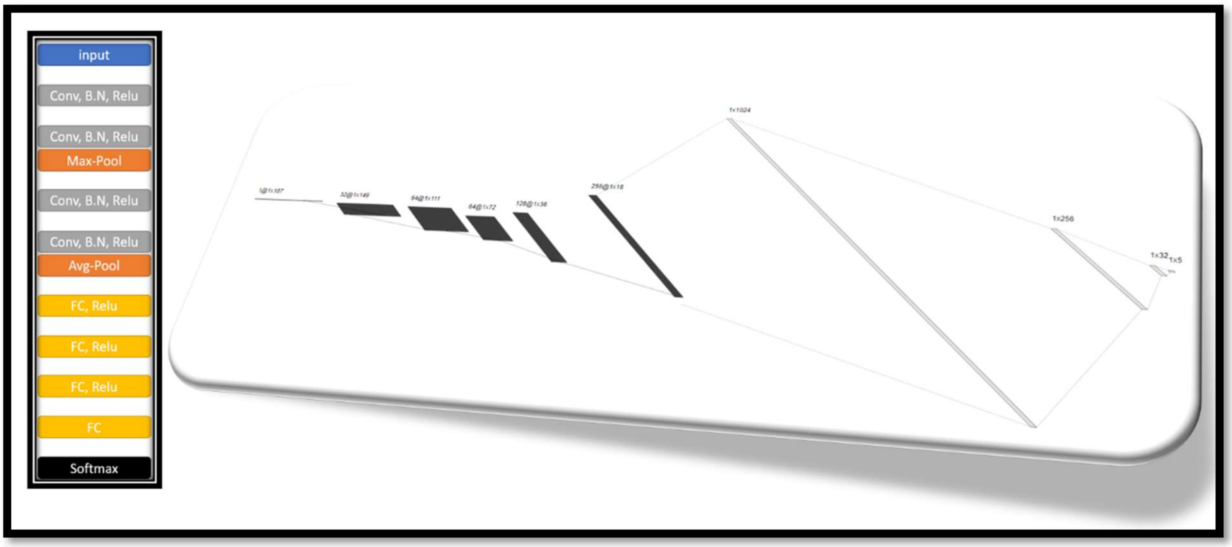


Figure 3: proposed model

And we compare results afterwards. So the model is based on 4 CN layers, then 4 FC layers and Softmax at last.

Number of trainable weights in the model: 5,125,125
Model size: 164004000 / bit | 20.50 / MB

We then prepared the model to be freezed to transfer into the other medical ECG based datasets.

Experiments and results (explanation of the dataset, the experiments you performed to validate your method and their results compared to previous method/s)

Experiments and results:

Datasets:

The Arrhythmia dataset consists of ECG recordings from 47 different subjects recorded at the sampling rate of 360Hz. Each beat is annotated by at least two cardiologists. We use annotations in this dataset to create five different beat categories in accordance with Association for the Advancement of Medical Instrumentation (AAMI) EC57 standard [18]. See Figure I for a summary of mappings between beat annotations in each category.

The MI (myocardial infraction) dataset consists of ECG records from 290 subjects: 148 diagnosed as MI , 52 healthy control, and the rest are diagnosed with 7 different disease. Each record contains ECG signals from 12 leads sampled at the frequency of 1000Hz.

In this dataset only ECG lead II included, and with MI and healthy control categories (binary).

Main features of datasets:

Arrhythmia Dataset

- Number of Samples: 109444
- train / test: 80% / 20%
- Number of Categories: 5
- Sampling Frequency: 125Hz
- Data Source: Physionet's MIT-BIH Arrhythmia Dataset
- Classes: ['N': 0, 'S': 1, 'V': 2, 'F': 3, 'Q': 4]
- Classes proportion: [0.83, 0.03, 0.07, 0.01, 0.07]

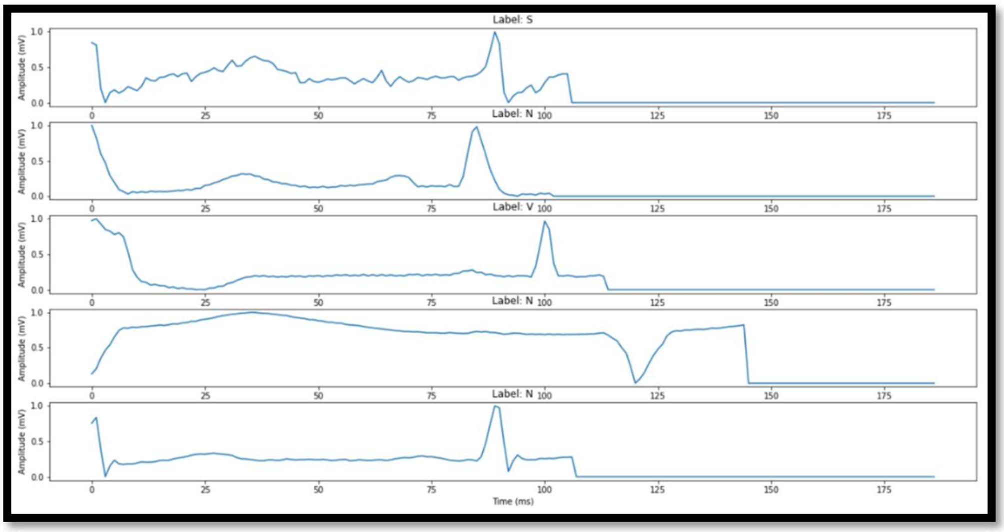
MI (myocardial infarction) Dataset

- Number of Samples: 14550
- train / test: 80% / 20%
- Number of Categories: 2
- Sampling Frequency: 125Hz
- Data Source: PTB Diagnostic ECG Database
- Classes: ['normal': 0, 'abnormal': 1]
- Classes proportion: [0.28, 0.72]

As one can immediately note, there is imbalance between the classes. in medical diagnosis, and particularly for ECG signals, data are highly imbalanced because samples exhibiting disease are more limited than normal samples. Dataset imbalance can cause two problems. First, the training becomes ineffective, as most observations are easy samples (normal samples) that provide no learning benefit to the model. Second, normal samples can dominate the training and cause the classifier to favor classes that have many labeled samples. Many address this problem with some augmentation, but we assume that augmentation distort the original data meaning in this case, based on Taissir 2020. So to address that, we performed train with this raw proportion, and by weighting the loss and accuracy, and calculating confusion matrices, we were able to measure the success of our model.

To demonstrate the signal we show 5 random signal from the Arrhythmia dataset with different labels:

Figure 4: 5 plotted ECG signal examples and their labels



The MI dataset contain similar time vectors ECG, but this time with labels "healthy" and "unhealthy."

So, the datasets are formally $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$ in which $\mathcal{X} \subseteq [0,1]^{187}$, and for all $x_i \in \mathcal{X}$, for the Arrhythmia dataset, $y_i \in \{0, ..4\}$, and for the MI dataset, $y_i \in \{0,1\}$.

Now we can address and define the transfer task:

Consider the Arrhythmia dataset as the source domain $\mathcal{D}_S$ and the MI task as the target domain $\mathcal{D}_T$.

We know that $\mathcal{X}_S = \mathcal{X}_T$, but $\mathbb{P}[\mathcal{X}_S] \neq \mathbb{P}[\mathcal{X}_T]$ and $\mathcal{Y}_T \neq \mathcal{Y}_S$ (beneath all of this lies another transfer learning which is augmentation – different distributions of $\mathbb{P}[\mathcal{Y}|\mathcal{X}]$).

Our freeze method is to transfer the model learnt for $\mathcal{D}_S$ and changing the k last FC layers to reach $\mathcal{D}_T$, for $1 \leq k \leq 4$. We assume that with very simple convolutional network, it is still very ECG representative, so the problem of $\mathbb{P}[\mathcal{X}_S] \neq \mathbb{P}[\mathcal{X}_T]$ and $\mathcal{Y}_T \neq \mathcal{Y}_S$ will be solved.

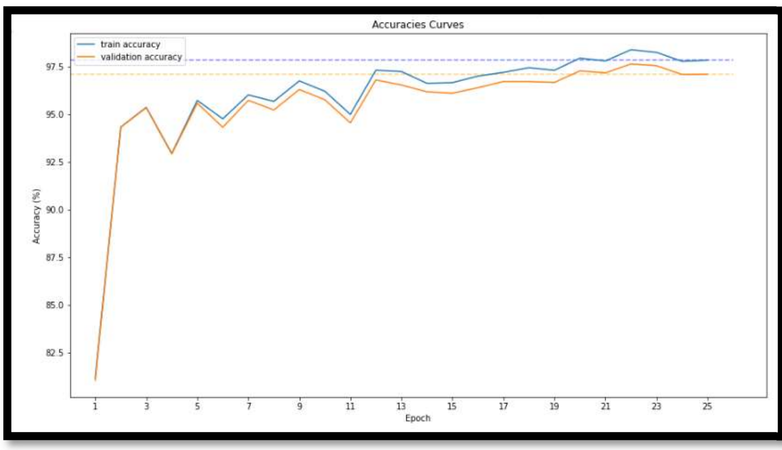
Experiments and results:

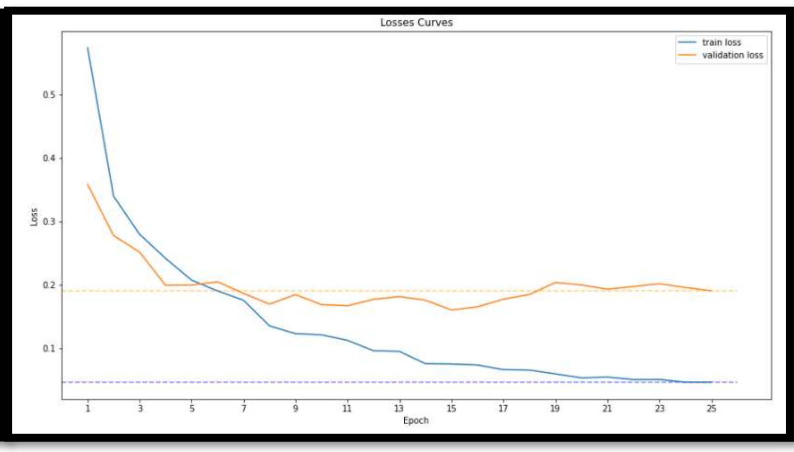
1) We first tried to train the model to the 2 datasets. Each one with early stopping and learning rate schedule. To do so, we trained the Arrhythmia dataset with 10 trials of optuna, on the hyperparameters:
   - $optimization \in \{Adam, RMSprop, SGD\}$
   - $lr \in range(1e^{-4}, 1e^{-2})$
   - $optimizer\ weight\ decay \in range(1e^{-5}, 1e^{-3})$
   - $batch\ size \in range(64,256)$

   The best set we got: $\{RMSprop, 3 \cdot 10^{-4}, 2 \cdot 10^{-4}, 87\}$

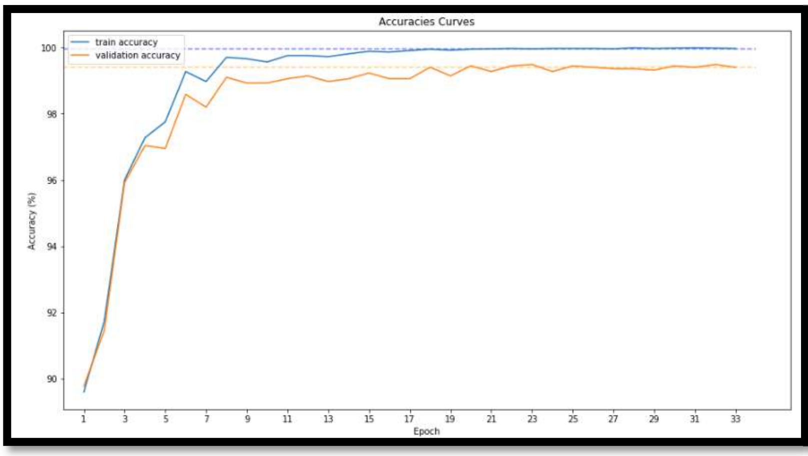
   We trained the 2 model afterwards with those hyperparameters and got these train curves:

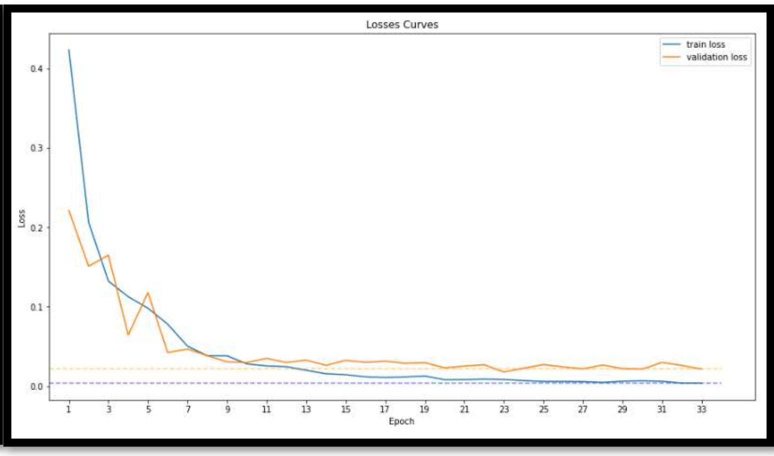
   For Arrhythmia dataset train:





For MI dataset train:

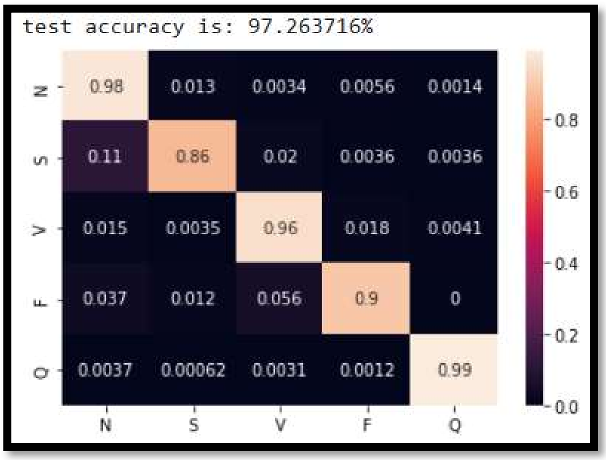


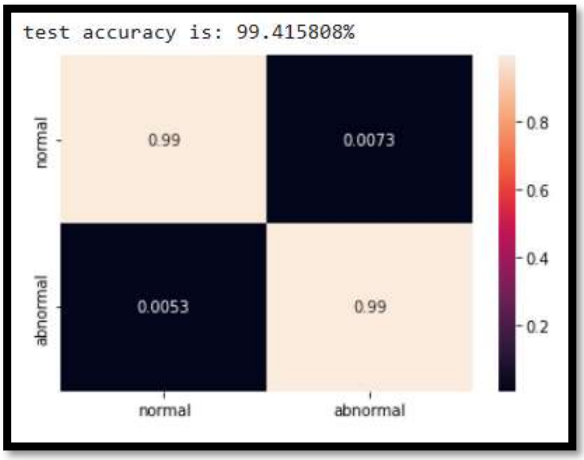


And we reached 97.26% and 99.41% test accuracy respectively.

If we calculating weighted accuracy (to balnce data) we reach 93.8% and 99.2% accuracy respectively.

With confusion matrices:





Confusion matrices for the classes with size proportion of:

Arrhythmia: [0.83,0.03,0.07,0.01,0.07]

MI: [0.28,0.72]

We can note from these that we alleviate the imbalance phenomenon without any augmentation, just by defining different loss, which weighting the classes i.e., with simple enough model compared to the state-of-the-art model proposed by Kachuee et AL. With all of this, the accuracy or weighted accuracy measurements, didn't fell from those aforementioned methods. *Important to mention that optuna get to conclusion that hyperparameters contribute the same to learning process. We show here a table which compare our results to others and their model complexity:

Table 1: model complexity – accuracy comparison

source	Model	Test accuracy (weighted by class size)	
		MIT-BIH	PTB
Kachuee et Al, 2018	CNN, 13 layers residuals	93.4%	95.9%
FekihRomdhane et Al, 2020	CNN, 14 layers residuals	98.6%	X
Sajiddeboss (on Kaggle)	CNN, 14 layers residuals	98%	X
Previous work (on Kaggle)	CNN, 8 layers	92.4%	99.9%
Ours	CNN, 8 layers	93.8%	99.21%

So, in the complexity ↔ accuracy dimension measurement, our model is found at a good place.

2) Transferability: Now we conducted 4 trials to show that we can transfer this simple model, trained on the Arrhythmia dataset to the MI dataset. In each $k \in \{1, ..4\}$ trial, we trained the last k fully connected layers of the Arrhythmia model on the MI datasets (of course with change of last layer to 2 classes). With the accuracy achieved, we can assume that the **convolution layers** encoded the data needed for the classification, which can be transferred between the tasks. To validate this we conducted several trials with **simple MNN** and got no good results, which implies on the **CN** impact on representation and separation of the data.
Moreover, we showed that due to early stopping mechanism, $parameter\ k$ matters a little for the whole training time (and even took more time for training than without transfer – due to convergence point noise), but significantly influences the accuracy, such that we can note that for $k = 4$, the accuracy is as high as without transferring.

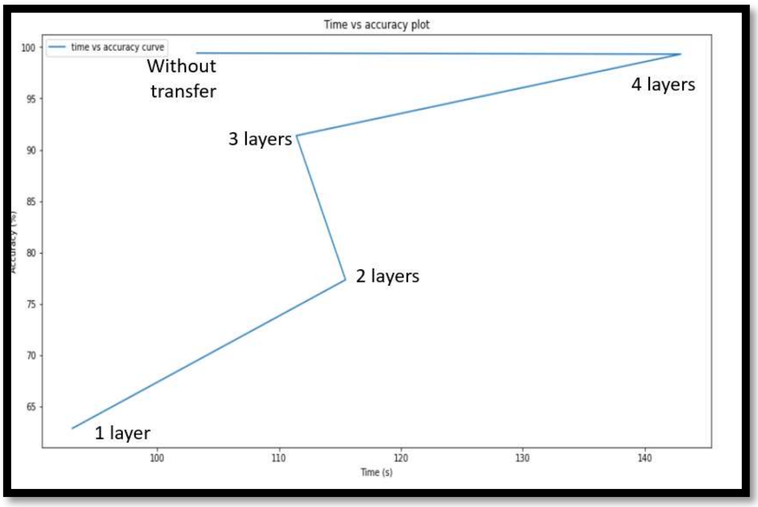
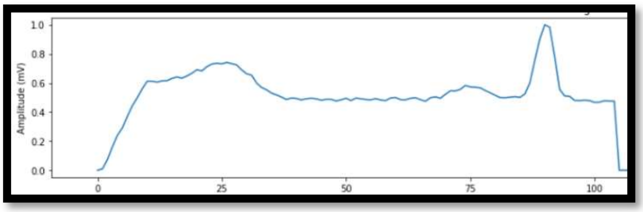
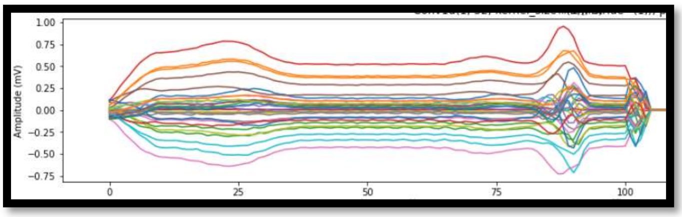


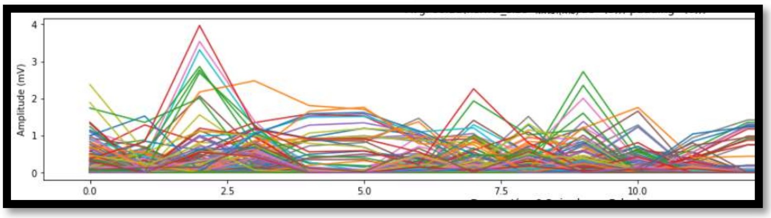
Figure 5: time vs accuracy plot for transfer FC training layers

3) Explainability: As we mentioned in the introduction, Tsne mapping performed on the last CN layer and results shown good separation w.r.t the Arrhythmia dataset. We wanted to illustrate and conclude the results by plotting the whole EGC signal layers by taking representative signals. The results imply that on deep layers, the representation based mainly on the initial signal “peaks” because, the other signal lowered down and the peaks bolded, but now each vector is at length 18 instead of 187 – lower dimension of peaks encoding vectors. The FC layer for example just

flattened the data to make a plot resembling some periodic wavelets, which emphasizes our conclusion that the data lies in lower dimension of peaks, and by encountering them, they define a wavelet, win which its frequency / amplitude or other characteristics defining it to one of the disease classes. This conclusion is well resembling the way physician classify the data, by the rhythm of peaks and periodicity. We show for example of ECG input, 2 CN layers:







<u>Main conclusions and further work:</u>

1) We conclude that dataset can be well separate using simple model, with good accuracies, and with referring to the minority class which is so important in the medical field. Moreover, the method doesn't require any assumption on data – from data preprocessing (Kachuee et Al, 2018) to some distorting augmentation.
2) The model is transferable from Arrhythmia task to the MI task, which preserve good accuracies based on FC training alone, that implies on ECG signal deep representation done by CN layers.
3) Maybe future work can be done on encoding the most important time values on each signal to a lower dimension vector, thus creating simpler but still expressive data. We assume that with self/unsupervised learning this can be done, thus building new pipeline to classify ECG signals.
4) It was interesting to check the conclusions on other real patients, because the Arrhythmia datasets is based on 47 patients which any conclusion may be much overfitting compared to the whole population. In addition to check more leads other than lead 2#.

<u>Acknowledgment</u>:

"**ECG Heartbeat Classification: A Deep Transferable Representation**" - Mohammad Kachuee et Al, 2018.

"Improved Neural Network Arrhythmia Classification Through Integrated Data Augmentation" - Garrett I. Cayce et Al, 2022.

"**Architecture Enhancement of Convolutional Neural Networks for Arrhythmia Classification"** - Hae Jin Kim et Al, 2022.

"Electrocardiogram heartbeat classification based on a deep convolutional neural network and focal loss" - Taissir FekihRomdhane et Al, 2020.

https://github.com/sajiddeboss/ECG-HeartBeat-Classification

https://www.kaggle.com/datasets/shayanfazeli/heartbeat