

---

# APPLYING DEEP-CLUSTERING APPROACHES TO CREATE MEANINGFUL REPRESENTATIONS OF ECG BEAT MORPHOLOGY

---

236781 - DEEP LEARNING ON COMPUTATIONAL ACCELERATORS  
FINAL PROJECT

**Amit Rotner**  
amitrotner@campus.technion.ac.il

**Shaked Doron**  
shaked.doron@campus.technion.ac.il

August 25, 2020

## ABSTRACT

In this project, we aim to research the effect of joint dimensionality reduction and clustering on an ECG diagnostic task. Our goal is to discover whether we can benefit from the latent space and clustering structures, to improve the model predictions. We compare unsupervised learning with dimensionality reduction and clustering, with the traditional supervised method. We experiment with both methods on the labeled PhysioNet MIT-BIH Arrhythmia and PTB Diagnostic ECG Databases [2, 8, 4]. **TODO:** insert key-results. The code is available at <https://github.com/amitrotner/CS236781-Project>

## 1 Introduction

ECG is the main tool for monitoring cardiac health. Today, ECG data are analyzed manually by physicians for detecting abnormalities in the signal. This manual procedure is both time-consuming and prone to errors. Therefore, we are trying to figure out a suitable method to tackle this problem and to increase the accuracy of detection and analysis. Hopefully, machine learning methods will make this manual process automatic in the near future.

In the field of machine learning, there are two main types of tasks: supervised, and unsupervised. In supervised learning, we map an input to output labels or to continuous output. On the other hand, in unsupervised learning, we wish to learn the inherent structure of our data without using explicitly-provided labels. While intuitively, ECG heartbeat classification is a supervised learning task, we think that a dimensionality reduction to a learned latent space can infer the natural structure of the ECG signals, leading to better cardiac health monitoring.

In this work, we use the DCC algorithm [10] for a joint nonlinear embedding and clustering procedure as a first approach and a convolutional neural network as a second approach. We compare those approaches, their drawbacks, advantages, results and performances on the PTB Diagnostic ECG Database.

## 2 Methods

### 2.1 Dataset

In this project, we use PhysioNet MIT-BIH Arrhythmia and PTB Diagnostic ECG Databases as a data source for labeled ECG records [2, 4]. The MIT-BIH Arrhythmia dataset contains records from 47 subjects. Those samples were recorded at the sampling rate of 360Hz while two or more cardiologists independently annotated each record. Those annotations are mapped into 5 different categories in accordance with Association for the Advancement of Medical Instrumentation (AAMI) EC57 standard [3]. The following table presents this mapping:

Category	Annotations
N	<ul style="list-style-type: none"> <li>• Normal</li> <li>• Left/Right bundle branch block</li> <li>• Atrial escape</li> <li>• Nodal escape</li> </ul>
S	<ul style="list-style-type: none"> <li>• Atrial premature</li> <li>• Aberrant atrial premature</li> <li>• Nodal premature</li> <li>• Supra-ventricular premature</li> </ul>
V	<ul style="list-style-type: none"> <li>• Premature ventricular contraction</li> <li>• Ventricular escape</li> </ul>
F	<ul style="list-style-type: none"> <li>• Fusion of ventricular and normal</li> </ul>
Q	<ul style="list-style-type: none"> <li>• Paced</li> <li>• Fusion of paced and normal</li> <li>• Unclassifiable</li> </ul>

The PTB Diagnostics dataset contains 549 records from 290 subjects. Within the header file of most of these ECG records, is a detailed clinical summary almost for every subject, except for 22 subjects whose clinical summary is not available. The diagnostic classes of the remaining 268 subjects are summarized in the table below:

Diagnostic class	Number of subjects
Myocardial infarction	148
Cardiomyopathy/Heart failure	18
Bundle branch block	15
Dysrhythmia	14
Myocardial hypertrophy	7
Valvular heart disease	6
Myocarditis	4
Miscellaneous	4
Healthy controls	52

For comparison purposes, we have only worked with MI and healthy control categories, the same as the baseline paper [5].

## 2.2 Baselines

Our presented methods are compared to Kachuee et al. [5], in which the authors proposed a transferable method based on deep convolutional neural networks for the classification of heartbeats.

In the following we will lay the mathematical background for understanding both approaches we are presenting. For the CNN see subsection 2.3 and for the DCC algorithm see subsection 2.4.

## 2.3 Convolutional Neural Network

Convolutional Neural Network (CNN) is one of the most commonly used types of artificial neural networks. As traditional neural networks use matrix multiplication as a way to connect between the input and the output, CNNs introduce convolutional layers as a replace for the general matrix multiplication. Convolutional layers convolve the input using their kernel. Given an  $m$ -dimensional vector  $x = (x_1, \dots, x_m)$ , the layer produces an  $n$ -dimensional vector  $y = (y_1, \dots, y_n)$  by the equation below:

$$y_j = \varphi \left( \sum_{i=1}^m w_{ij} * x_i \right) \quad (1)$$

where  $\varphi$  is an activation function and  $*$  is the convolution operator.

Moreover, CNNs introduce pooling layers (or subsampling layers) which are a way to reduce the size of feature maps between the convolutional layers. For instance, the maximum pooling uses the maximum value in the adjacent area as an output and the average pooling uses the average.

For further readings about CNNs, see the relevant pages at the course website: 1 & 2

## 2.4 DCC algorithm

(This subsection is heavily based on [10])

Let  $X = [x_1, x_2, \dots, x_N]$  be a set of points in  $\mathbb{R}^D$  to be clustered. Clustering algorithms rely on inter-point distances. Hence, when  $D$  is high, these distances become less informative leading to a non-effective clustering [1]. To overcome this problem, we embed the data into a lower dimensional space  $\mathbb{R}^d$ . The embedding of the dataset into  $\mathbb{R}^d$  is denoted by  $Y = [y_1, y_2, \dots, y_N]$ . The function that performs the embedding is denoted by  $f_\theta : \mathbb{R}^D \rightarrow \mathbb{R}^d$ . Thus, we can summarize that  $y_i = f_\theta(x_i)$  for all  $i$ .

To constrain  $f_\theta$  to construct a faithful embedding of the original data, we require that the original data be reproducible from its low-dimensional image. Therefore, we consider a reverse mapping  $g_\omega : \mathbb{R}^d \rightarrow \mathbb{R}^D$  and minimizing the following reconstruction loss:

$$\min_{\Omega} \|X - G_\omega(Y)\|_F^2 \quad (2)$$

where  $\Omega = \{\theta, \omega\}$ ,  $Y = F_\theta(X) = [f_\theta(x_1), \dots, f_\theta(x_N)]$ ,  $G_\omega(Y) = [g_\omega(y_1), \dots, g_\omega(y_N)]$  and  $\|\cdot\|_F$  is the Frobenius norm.

In order to do a clustering in tandem with continuous optimization of the embedding, the Deep Continuous Clustering (DCC) algorithm is built on the RCC formulation [9]. The RCC formulation introduces the idea of using a set of representatives  $Z \in \mathbb{R}^{d \times N}$ , initialized at the corresponding latent data points  $Y$  and optimizes the following objective:

$$\min_Z \frac{1}{2} \|Z - Y\|_F^2 + \frac{\lambda}{2} \sum_{(i,j) \in \mathcal{E}} w_{i,j} \rho(\|z_i - z_j\|_2) \quad (3)$$

where the function  $\rho(\cdot)$  is a penalty on the regularization terms,  $\mathcal{E}$  is the set of edges in a graph connecting the data points. The latter is constructed automatically from the data. The first term in 3 ensures that the representatives  $Z$  will be near the corresponding datapoints and the second term in 3 encourage the representative to merge. The weights  $\{w_{i,j}\}$  balance the contribution of each data point to the pairwise term and  $\lambda$  balances the two objective terms.

Influenced by the above, the DCC algorithm optimizes the following objective:

$$\mathcal{L}(\Omega, Z) = \frac{1}{D} \underbrace{\|X - G_\omega(Y)\|_F^2}_{\text{reconstruction loss}} + \frac{1}{d} \left( \underbrace{\sum_i \rho_1(\|z_i - y_i\|_2; \mu_1)}_{\text{data loss}} + \lambda \underbrace{\sum_{(i,j) \in \mathcal{E}} w_{i,j} \rho_2(\|z_i - z_j\|_2; \mu_2)}_{\text{pairwise loss}} \right) \quad (4)$$

where  $Y = F_\theta(X)$ .

## 3 Implementation and experiments

### 3.1 Preprocessing

To extract beats from the ECG signals, the data were preprocessed using the technique presented in [5]. The extracted beats have the same dimension which is essential for training the models.

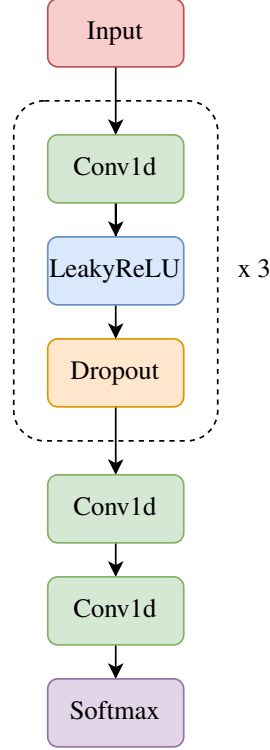


Figure 1: CNN Model Architecture

### 3.2 Training the CNN network

The figure 1 illustrates the network architecture proposed for both datasets. The model inputs, as explained in 3.1, are extracted beats with a dimension of 187. All convolution layers are applying 1-D convolution through time and each have a kernel of size 3, a stride of size 1 and a padding of size 1. After each 1-D convolution we used a LeakyReLU layer [7] with negative slope of 0.2 and a dropout layer [11] with a probability of 0.1. The last two 1-D convolutions function as a fully-connected layers followed by a Softmax layer to produce class probabilities. In total, the network consists 5 weight layers.

### 3.3 Training the DCC network

TODO

### 3.4 Implementation Details

We have used PyTorch computational library for training and evaluating the models. In the CNN models training, we have used Cross Entropy loss as the loss function, while during the DCC model training, we have used the loss function developed in 2.4. For both methods training, we have used Adam optimizer [6] with a learning rate, beta-1, beta-2 and Epsilon of 0.001, 0.9, 0.999 and 1e-8, respectively. The learning rate is decayed exponentially with the decay factor of 0.75 every 500 iterations. We have the models for 3000 epochs, with an early stopping of 5 epochs without improvement. Those hyperparameters were taken from the baseline paper [5], with some small modifications, for reliable comparison. We have trained all the networks in the Rishon server, on a GeForce RTX 2080 processor.

## 4 Results and discussion

We have evaluated the methods on a test set contains 20% of the data which is not used in the training phase. Tables 4 and 4 show the results of PTB/MIT-BIH dataset, respectively.

Comparison of PTB dataset results

Work	Accuracy
<b>Ours CNN network</b>	<b>96.7%</b>
<b>Ours DCC network</b>	<b>TODO</b>
Baseline paper [5]	95.9%

Comparison of MIT-BIH dataset results

Work	Accuracy
<b>Ours CNN network</b>	<b>92.7%</b>
<b>Ours DCC network</b>	<b>TODO</b>
Baseline paper [5]	93.4%

## References

- [1] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? *ICDT 1999. LNCS*, 1540, 12 1997.
- [2] Kreiseler D. Bousseljot R. and Schnabel A. Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet. *Biomedizinische Technik / Biomedical Engineering*, 40, 1995-01-01.
- [3] Association for the Advancement of Medical Instrumentation and American National Standards Institute. *Testing and Reporting Performance Results of Cardiac Rhythm and ST-segment Measurement Algorithms*. ANSI/AAMI. The Association, 1999. ISBN 9781570201165. URL <https://books.google.co.il/books?id=gzPdtgAACAAJ>.
- [4] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13). Circulation Electronic Pages: <http://circ.ahajournals.org/content/101/23/e215.full> PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.
- [5] Mohammad Kachuee, Shayan Fazeli, and Majid Sarrafzadeh. ECG heartbeat classification: A deep transferable representation. *CoRR*, abs/1805.00794, 2018. URL <http://arxiv.org/abs/1805.00794>.
- [6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.
- [7] Andrew L. Maas. Rectifier nonlinearities improve neural network acoustic models. 2013.
- [8] G. B. Moody and R. G. Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.
- [9] Sohil Shah and Vladlen Koltun. Robust continuous clustering. *Proceedings of the National Academy of Sciences*, 114:201700770, 08 2017. doi: 10.1073/pnas.1700770114.
- [10] Sohil Atul Shah and Vladlen Koltun. Deep continuous clustering. *arXiv:1803.01449*, 2018.
- [11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL <http://jmlr.org/papers/v15/srivastava14a.html>.