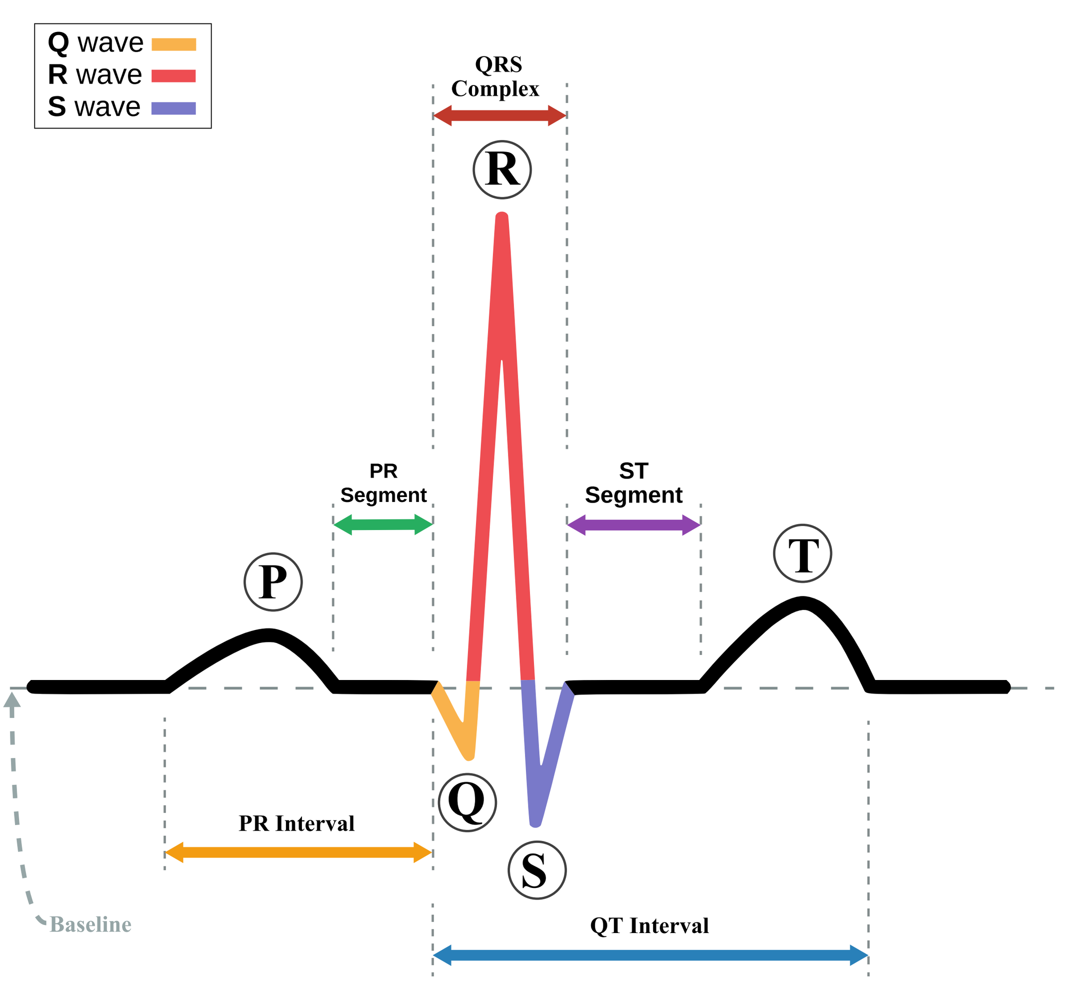
Motivation

With the rapid development of artificial intelligence and new and exciting machine learning methods, we can train computers to process large amounts of data very quickly. Today we can automate important procedures, such as, fault diagnosis in equipment, or in fact humans. We can use these tools to monitor our health and in this project I studied **Electrodiagrams (ECG)**, to monitor the proper functioning of the cardiovascular system. With these methods professionals can save some time from scattering endless ECG results and focus on things only humans can do.

This was my first project last year, but I could not achieve better results with the wavelet transformation so I thought I give it a second try.



Introduction and background

Introduction

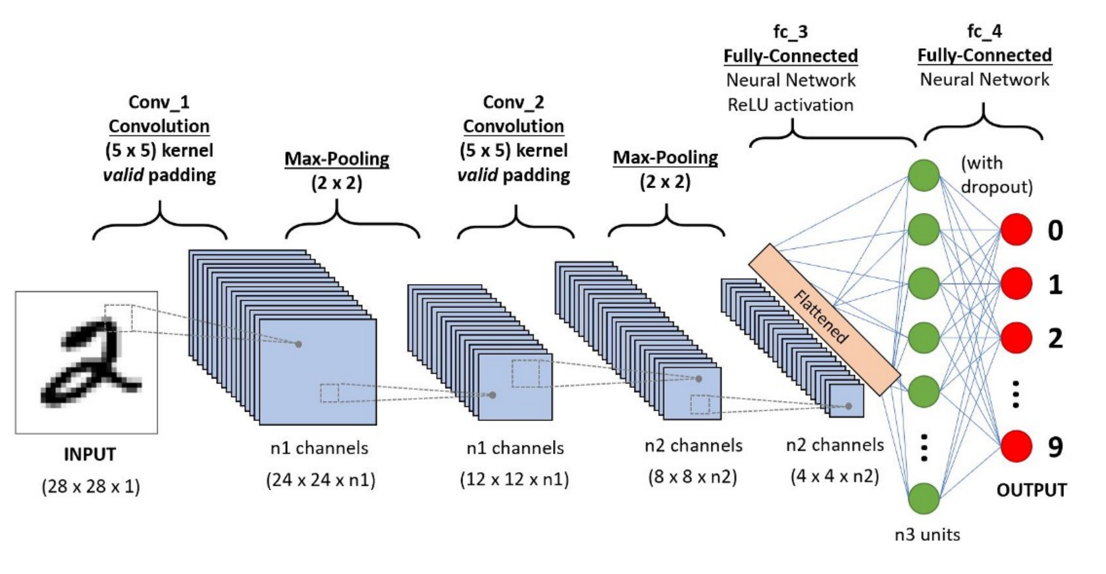
Scientists achieved excellent results working with machine learning in the classification of ECG signals (ECG Heartbeat Classification: A Deep Transferable Representation, Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation). They can classify the different ECG heartbeat classes:

- the normal beat [**N**] *left bundle branch block beat [**L**]*right bundle branch block beat [**R**] *atrial premature beat [**A**]*ventricular beat [**V**]

to more than 90 percent accuracy. In this project I will try to recreate these results with Neural Networks and signal processing methods.

Neural Networks

Artificial neural networks are machine learning methods, it is a circuit of artificial neurons, which communicate information between each other, inspired by biological neural networks, mimicking the connections between them with weights corresponding to each node (neuron). They can be trained to form probability-weighted associations between input and result. This can be achieved by feeding it enough examples of the certain problem, and then the model adjusts to the criteria. This is called supervised learning.



The neural network I choose to work with is a **Convolutional Neural Network (CNN)**. These are fully connected networks, that uses the method of convolution to scan through the data, typically an image and give them weights and pass them to the next layer. The full connectivity means each node in one layer is connected to every node in the next. This means the CNN is prone to overfitting, which means it knows the training data to well and can't generalize. To help regularize this, we use penalties, such as dropping a chunk of the data before the next layer to prevent overfitting.

Wavelet Transform

The signal processing part of my project is the implementation of the wavelet transform. It is similar to the Fouirer transform, but it uses wavelets instead of sine waves to analyze dynamical frequency spectrums. It has high resolution in both the frequency and time domain which means, it can tell which frequencies are present in the signal and which time they occurred. Wavelet transforms use a process called convolution, which means we move our original (mother) wavelet through the signal step-by-step. Then we can scale the wavelet and repeat. In the project I use the **continuous wavelet transform (CWT)** of the ECG signals to make a scaleogram which I can use to train a 2D convolutional neural network. The Wavelet transform of a 1-D signal have two dimensions. The formula of the CWT is:

$$X_w(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t)\psi^*\left(\frac{t-b}{a}\right)\, dt$$

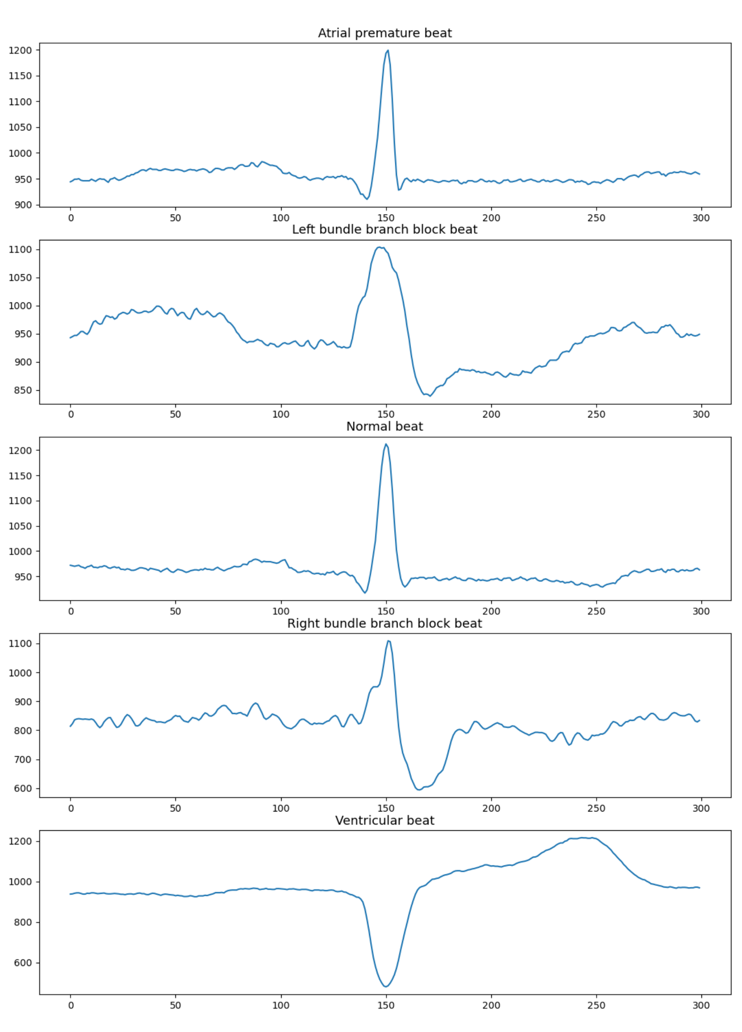
where ψ is the continuous mother wavelet that gets scaled by the a factor and translated by the b factor. These values are continuous, so there can be infinite amounts of wavelets.

Data

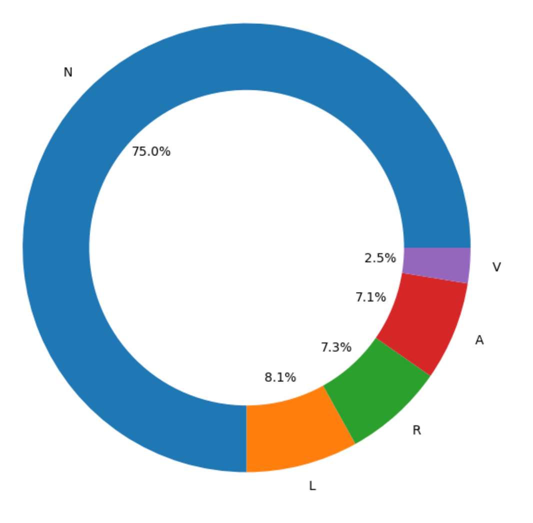
The dataset I used is the MIT-BIH Arrhythmia Database from physionet and I got a friendlier format from Kaggle. For the wavelet transformation I used the Pywavelets package . The neural networks were made with Keras and TensorFlow.

Preprocessing

The data has 48 subjects ECG signals and the corresponding annotations, which contain information about the class of each heartbeat. I linked the correct labels to the corresponding heartbeats and placed them in a big dataframe. This is how the different heartbeats look.

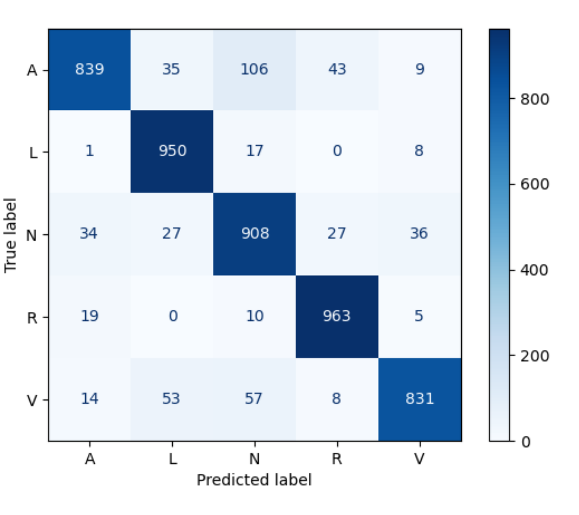


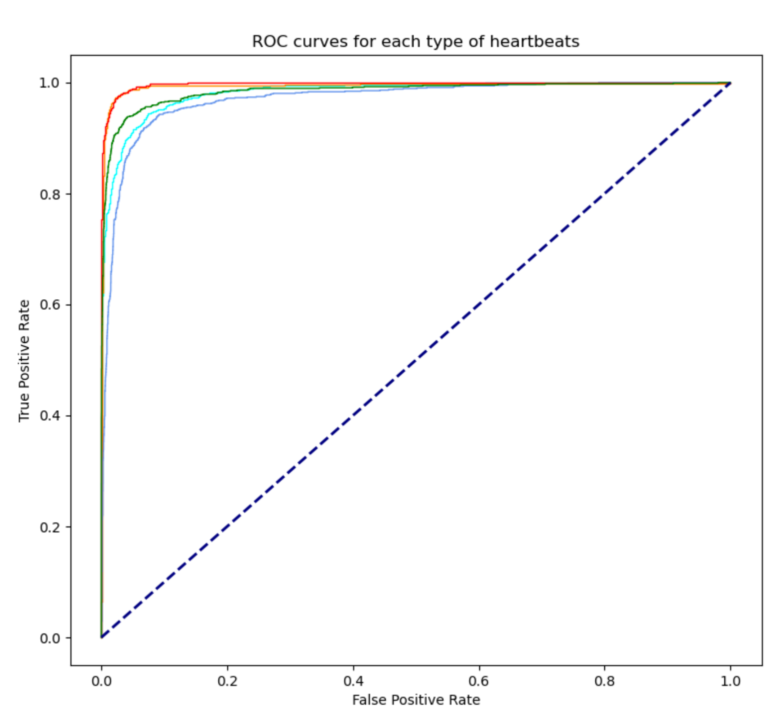
It was clear that the majority of the heartbeats were normal, so I downsampled the normal class and resampled the other ones.



Model 1

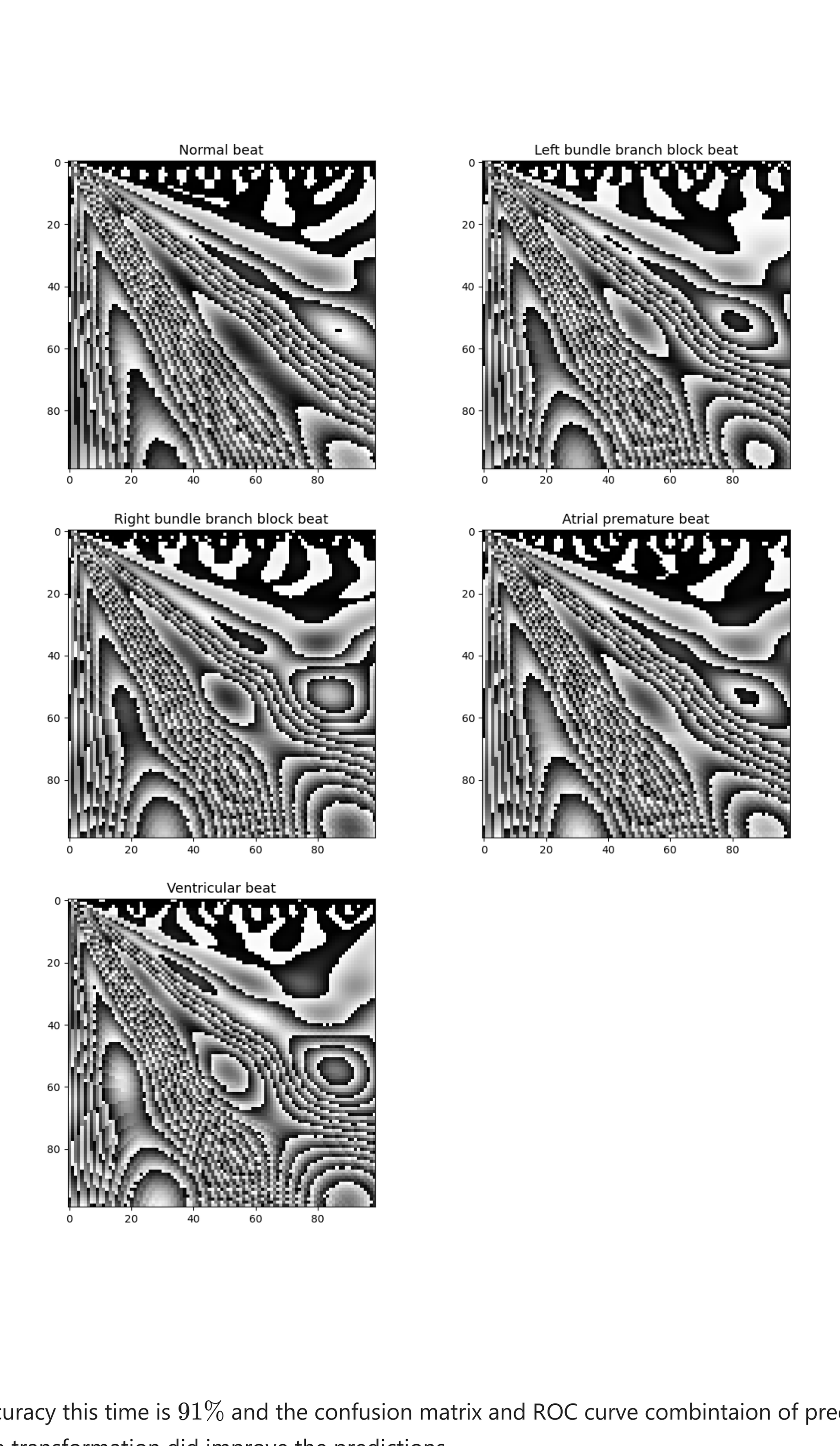
The next step was to prepare the dataset to the training of the CNN model, so I used the train-test-split method to make a training and testing set from the data. I switched the labels to numerical classes, in order to use them as the training and testing label set with the CNN model. The model contains Conv1d and MaxPooling layers in my model, because they can excecute convolution on 1D data, then I used a Dense layer to make the final output predictions. The model achieved 89% accuracy on the testing data, which means it can accurately predict the class of a signal the model never seen before. We can see the confusion matrix and ROC curve below





Model 2

In the articles cited above, the writes could make a model that performs better than the model with the Conv1D layers. They took the Continuous wavelet transform of the signals and made a scaleogram of them and trained a 2D convolutional neural network model with them. I tried to replecate this, and this time it went better. The first model did very well also, but this time the 2D convolutional data was also accurate. Below you can see images from each class after the CWT.



The accuracy this time is 91% and the confusion matrix and ROC curve combintaion of predictors look good also, so we can conclude, that the transformation did improve the predictions.