Heart rhythm interpretation using a convolutional neural network

ELE690 project

Sindre Mikkelsen (256447)
Halvor Kvamme (253683)
Malin Harr Overland (254690)

Department of Electrical Engineering and Computer Science
University of Stavanger
Stavanger, Norway

November 29, 2023

Abstract

This project developes a convolutional neural network (CNN) to classify hearth rhythms in five different categories, ventricular fibrillation (VF), ventricular tachycardia (VT), asystole (AS), pulseless electrical activity (PEA), and pulse generating rhythm (PGR). The project was based around earlier projects from the university, and also took inspiration from studies outside of this. The necessary background to understand the project is presented, and a detailed description of both the dataset and the model that was made follows. The main purpose of the experiments was to find hyper parameters that would increase the accuracy of the model, these are presented in the results section. Lastly it was concluded that this model is generally a good fit for classifying hearth rhythms, and some suggestions for further improvement were made. The code and a description of how it works can be found in the git repository hosted at GitHub.

Contents

1 Introduction

Cardiac arrest is when the heart suddenly stops pumping, and this is one of the most common causes of death worldwide [15]. Out-of-Hospital cardiac arrest (OHCA) also happens to around 3000 people in Norway every year, and when this happens it would be useful with a machine that can tell you what is going on with the heartbeat and when to shock the person in cardiac arrest [16].

Between 2002 and 2004 there was conducted a study to measure the quality of CPR in three different cities, Akershus, Stockholm and London [17]. The study is called "Quality of Cardiopulmonary Resuscitation During Out-of-Hospital Cardiac Arrest", and this study collected ECG-data that was annotated by experts.

In 2022 the same dataset was used for a project at UiS [9]. The project built a neural network to classify the dataset into five classes, with the goal "to create a model that provides the best accuracy". The classes are explained in subsection 2.2. This years project builds on the code from last year, with additional features that were inspired by the article "Fully Convolutional Deep Neural Networks with Optimized Hyperparameters for Detection of Shockable and Non-Shockable Rhythms" [10]. The goal was to improve the network to get a better accuracy through more extensive testing and experimenting on the different hyper parameters.

2 Background

This section will explain how a convolutional neural network works in general, and what the different hyper parameters are. The hearth rhythms that are the five classes in the network will also be explained.

2.1 Convolutional neural network (CNN)

Convolutional Neural Networks (CNNs) are a class of feed-forward neural network that tries to extract features directly from data via filters. CNNs are particularly useful for finding patterns in structured grid data, such as images, audio, time-series, and signal data. This can be used for tasks like image classification, object detection, image segmentation, and signal classification, as is the case for this project. What makes a neural network a *convolutional* neural network is that at least one of the layers uses convolution instead of general matrix multiplication [8].

Convolution is a mathematical operation that combines two signals into a new third signal. The original signal is convolved with a filter, resulting in a new filtered signal. Typically for CNNs the kernels are much smaller than the input, which leads to sparse interactions between input and output units. This makes the model more effective and reduces the memory requirements [8]. The convolutional layers are the layers that extract meaningful features from the input data.

Another common layer in a CNN is the pooling layer. This is a downsampling layer which is used to reduce the spatial dimensions of the feature map after the convolutional layers. One common pooling method, and the one which is used in this project, is max pooling. In max pooling, the maximum value in each local region of the input feature map is found and used, while the rest of the values are discarded. Downsampling like this helps to decrease the computational load, while keeping essential information.

2.2 Heart rhythms

The heart rhythms in the dataset are divided into two main classes and five sub-classes. The main classes are shockable and non-shockable rhythm, which indicates whether a patient should or should not be shocked with a defibrillator. The sub-classes in the shockable class are ventricular fibrillation (VF) and ventricular tachycardia (VT). For the non-shockable class the sub-classes are asystole (AS), pulseless electrical activity (PEA) and pulse generating rhythm (PGR). A description of the sub-classes is listed below.

- VF: Ventricular fibrillation is an arrhythmia that affects the ventricles in the heart [13]. It means that the heart muscle quivers or twitches (fibrillates) instead of completely expanding and squeezing. This leads to the heart not pumping out blood as it should.
- VT: Ventricular tachycardia is another type of arrhythmia, marked by an uncharacteristically fast heartbeat [6]. Instead of beating 60-100 times per minute (as is normal), a heart that is in VT beats over a 100 times or more per minute. VT is caused by irregular electrical impulses in the ventricles, and can cause a sudden cardiac arrest or lead to the organs not getting enough oxygen.
- AS: Asystole is when the heart stops pumping entirely and there are no electrical signals, making it a type of cardiac arrest [5]. It is also often called "flat-line" or "flat lining", since it looks like a flat line on an electrocardiogram. The heart has no detectable electrical activity.
- PEA: Pulseless electrical activity is when the electrical activity in the heart is too weak to make it pump [4]. This means there are electrical signals there, but they do not make the heart contract. If it is not treated immediately the heart will go into asystole. The difference from AS is that with PEA there is still *some* electrical activity, although not enough to make the heart pump.
- PGR: Pulse generating rhythm is simply a heart rhythm that generates a pulse. This includes normal sinus rhythm, but is not limited to it [3].

3 Materials and methods

This section will go through the structure of the dataset, the model used in the experiments for classifying and the structure of the experiments that was done.

3.1 Dataset description

As mentioned earlier the dataset was collected for the study in [17]. There was collected ECG-data from 298 patients, and these were sorted into the five categories described in section 2.2. There are two different datasets within the data, one with compressions and one without. The dataset without compressions is called "Clean Cuts" and has 2833 cuts of ECG-data that was used for this project. The raw data was sampled at 500 Hz and with 16 bit resolution. For this project only the "Clean Cuts" dataset was used.

Figure 1 shows three samples from each category, where each cut spans four seconds. As seen the amplitude is different between the classes and therefore the samples were not normalized.

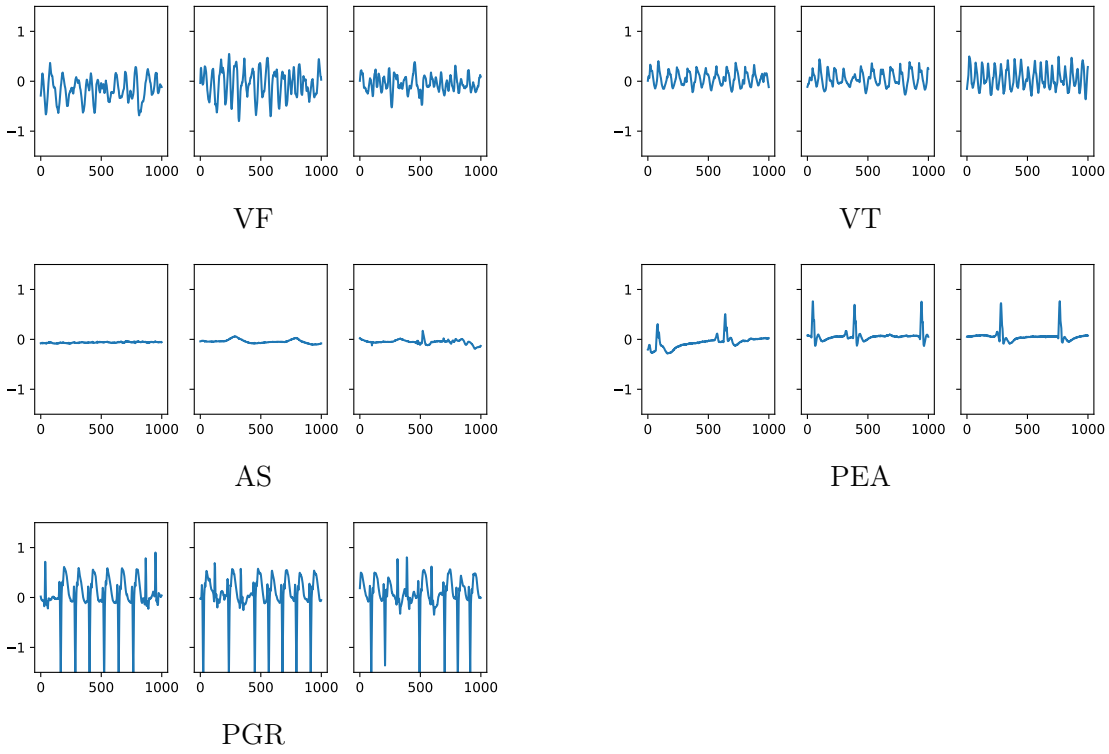

Figure 1: Three samples of the ECG signal for each category.

Figure 2 shows how many samples there are per category. It shows that the dataset is not balanced, as the biggest category, PEA (912 samples), has over 5 times as many samples as the

smallest category, VT (166 samples). Because of this, balanced accuracy was used to evaluate the models instead of normal accuracy. This is described more in subsection 3.4.

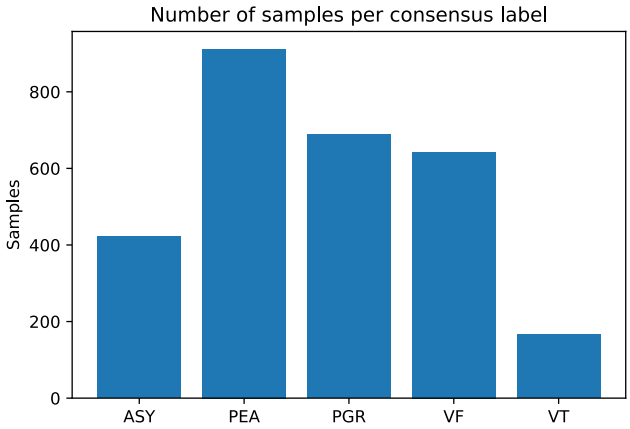

Figure 2: Number of samples per category in the dataset.

3.2 Dataset partitioning

The dataset was divided into training, validation and test-sets, first with 90% in the training set and 10% in the test set. Then the training set is split into ten folds using stratified k-fold cross validation to get validation and test data. The fact that it is stratified means that the ten folds has the same class distribution as the original dataset. For each trial, the weights in the model that gives the lowest validation loss will be saved and used to test against the test set.

3.3 Model architecture

Last years project, which was used as a starting point this year, tried out different structured models to find the one that worked best for classifying hearth rhythms [9]. This year, there was more focus on trying out different parameters for one general structure, and this was inspired by the article [10].

The CNN model is illustrated in figure 3. After the input layer there was implemented N convolutional blocks, where N is a tuneable parameter. Inside this convolutional block there are four different types of layers: 1D convolution, batch normalization, max pooling and a dropout layer. One-dimensional convolutional layers and max-pooling are both explained in subsection 2.1. The main purpose of the convolutional layer is to let the model learn some important features of the input data, while max-pooling decreases the spatial dimensions of the feature map.

Batch normalization is a technique used to normalize the inputs of a layer in a neural network. The main purpose of batch normalization is to improve optimization, and it helps to stabilize and accelerate the training process [8]. Lastly in this block there is a dropout layer. The purpose of a dropout layer is to prevent overfitting in the network. It chooses random neurons and sets them to zero with a certain probability.

All the layers in the model, except the output layer, uses the same activation function: Rectified Linear Unit (ReLU). ReLU has become the default activation function for CNNs, because of the good performance it achieves [12]. The output ReLU activation function is either directly the input if it is positive, and otherwise it is zero.

After the N convolutional blocks there comes a one-dimensional max-pooling layer, and then two dense layers. All these layers use ReLU as an activation function. The global max-pooling layer moves a window over the input, and outputs the maximum value for each window position. In a dense layer, all the inputs are connected to all the outputs, or in other words: the layer is fully connected.

Lastly comes the fully connected output layer, where the hearth rhythm is classified into one of the five hearth rhythms. Here the activation function that is used is called softmax. This activation function scales the output numbers into probabilities. This makes the output the probability for the heart rhythm to belong to the class.

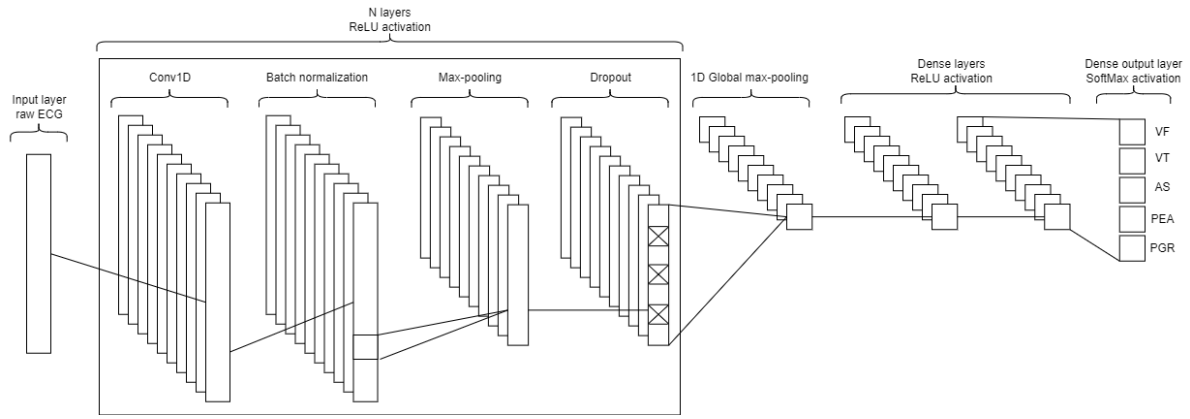

Figure 3: The structure of the CNN model.

3.4 Experiments

When testing the model some of the parameters were set to a fixed value. The batch size was set to 32, learning rate was set to 0.001 and there was a maximum of 250 epochs, with early stopping if the validation loss did not improve for 30 epochs. Additionally the number of fully connected hidden layers was set to two (as described in subsection 3.3), and the downsampling was done with the pool size set to two.

To test the effect of the different hyperparameters *Keras Tuner* was used to perform random search [14]. Keras Tuner is a framework with algorithms to find the best parameters for a model. For each N there was done 250 trials, with 10-fold cross validation in each trial. Every trial consist of randomly selected parameters. The parameters which were experimented with are listed below, with $i = 1, 2, .., N$ and $j = 1, 2$.

- N = [2, 3, 4, 5], layers of convolutional blocks in the model.
- Dropout: [0.1, 0.2, .., 0.9]
- K_i = [5, 10, 15, 20, 25, 30, 40, 50], kernel sizes in Conv1D
- F_i = [5, 10, 15, 20, 25, 30, 40, 50], number of filters in Conv1D
- FC_j = [16, 32, 64, 128], number of nodes in the fully connected hidden layers

We used Balanced Accuracy (BAC) as our criteria to evaluate the models since the dataset is very unbalanced. Balanced accuracy is a development of the normal accuracy, instead of looking at only how many of the predictions that are correct it looks at how many are correct for each class. This means that it is more accurate for imbalanced datasets [2].

$$\text{BAC} = \frac{\text{Sensitivity} + \text{Specificity}}{2}$$

After the initial experiments, the dropout and FC_j was fixed, and more experiments were carried out while varying F_i and K_i. This time there was done 400 trials for each N.

4 Results

The results section is divided in three parts. First the results of the random search without fixed parameters are presented, and then the results with two more of the parameters fixed are presented. At the end of each of these two subsections there is a table that shows the parameters for the most optimal models, and then confusion matrices to summarize the performance of the model. At last the results from the model giving the highest BAC from the trials is presented, where the results from another 10-fold cross validation training is presented to look at robustness and variation within the model.

4.1 Without fixed parameters

The scatter plot in figure 4 shows the relation between the accuracy and the number of trainable parameters in the model. The plot to the left uses the accuracy and the plot on the right uses the balanced accuracy. For the rest of the report the results are presented with regards to the BAC.

From the scatter plot it can be seen that the model does not necessarily improve with added complexity. The BAC does not improve noticeably with more trainable parameters. Since there are five classes, a $BAC = 20\%(= \frac{100}{5})$ equals the probability of correctly classifying by random guessing. For $N = 5$ there are some number of trainable parameters that makes the BAC go to 20%. Why this happens is explained more in the next paragraph. It is not possible to deduce which N works best from this scatter plot, but it is clear that $N = 2$ achieves poorer results than the rest.

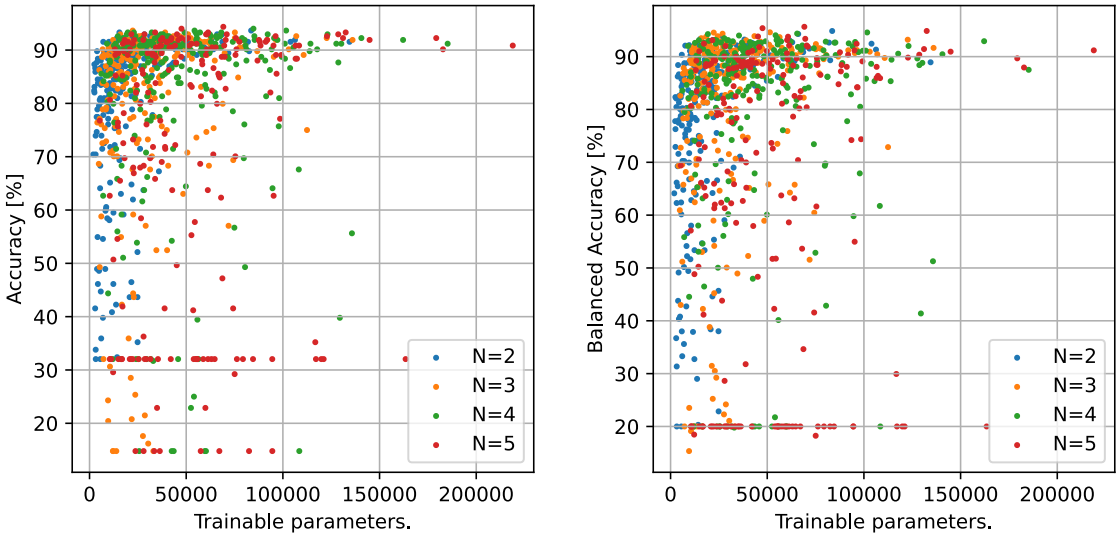

Figure 4: Scatter plots showing the relationship between number of tuneable parameters and the accuracy (on the left) or the balanced accuracy (on the right), for each N.

Figure 5 shows the BAC in relation to the dropout for N equal to two, three, four and five. The common factor in all four plots is that a too high dropout value results in a low accuracy. This is what causes the previously mentioned 20% BAC in the scatterplot. For $N = 5$ in figure 5 a dropout between 0.7 and 0.9 gives this accuracy of 20%. When the dropout value gets too high, too many neurons are set to zero, and the model will not have enough information. For all four N the best accuracy is achieved when the dropout median is 0.3.

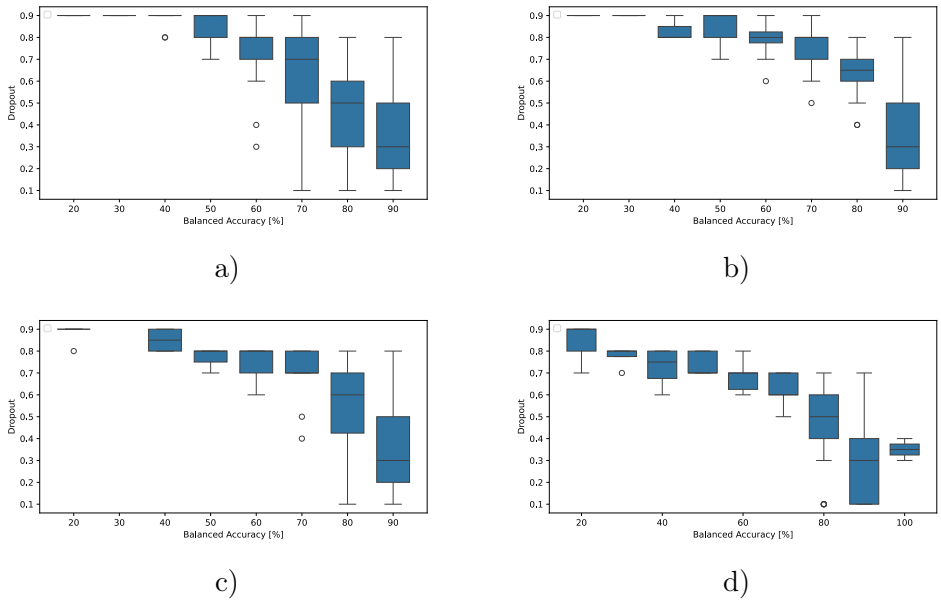

Figure 5: The relationship between dropout and BAC. For a) $N = 2$, b) $N = 3$, c) $N = 4$ and d) $N = 5$.

The third parameter that was experimented with was the kernel sizes, K_i. Figure 6 shows the relation between kernel size and the accuracy, with one block for each K_i. For $N = 2$ there seems to be a positive correlation between the two, so that a bigger kernel size yields a better accuracy. For $N > 2$ the relation is not as obvious. The kernel size's effect are tested further in the next subsection.

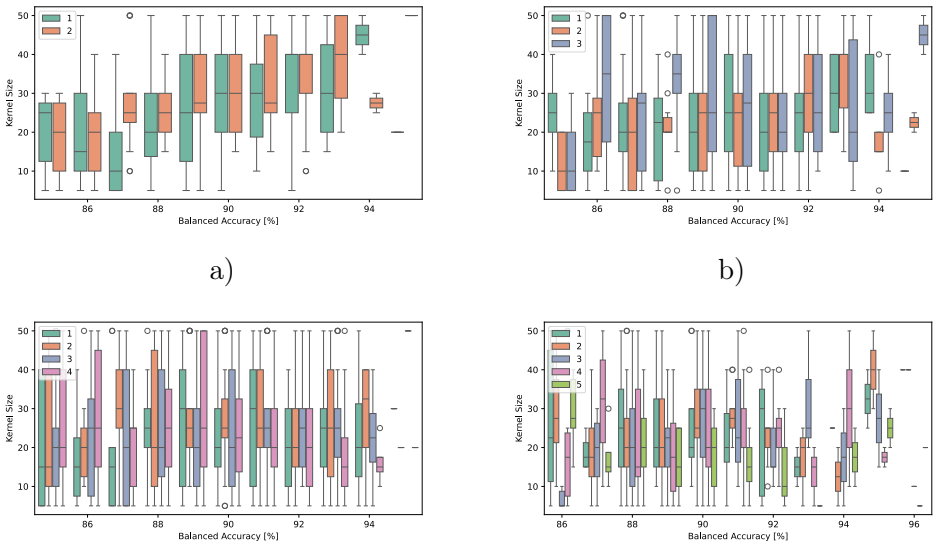

Figure 6: The relationship between kernel size and BAC. For a) $N = 2$, b) $N = 3$, c) $N = 4$ and d) $N = 5$.

For the number of filters, F_i, the relation with the accuracy is shown in figure 7. How these two relates to each other is not very obvious from the plots. This parameter was also tested further in the next subsection.

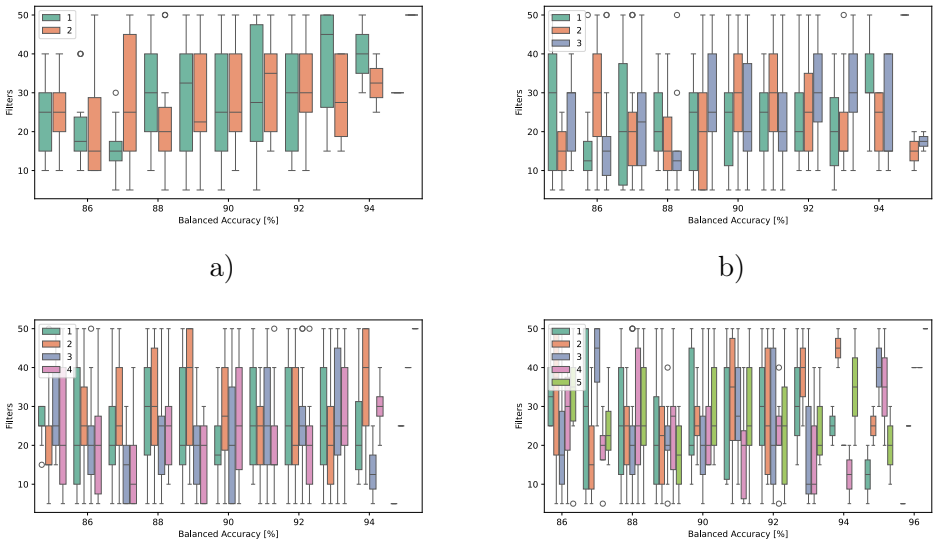

Figure 7: The relationship between number of filters and BAC. For a) $N = 2$, b) $N = 3$, c) $N = 4$ and d) $N = 5$.

Before the output layers there are two fully connected layers, and the size of these were the last parameter that was experimented with. Figure 8 shows the relationship between this size and the accuracy. From this it is possible to take the values with the best accuracy for each N, and this is summarized below.

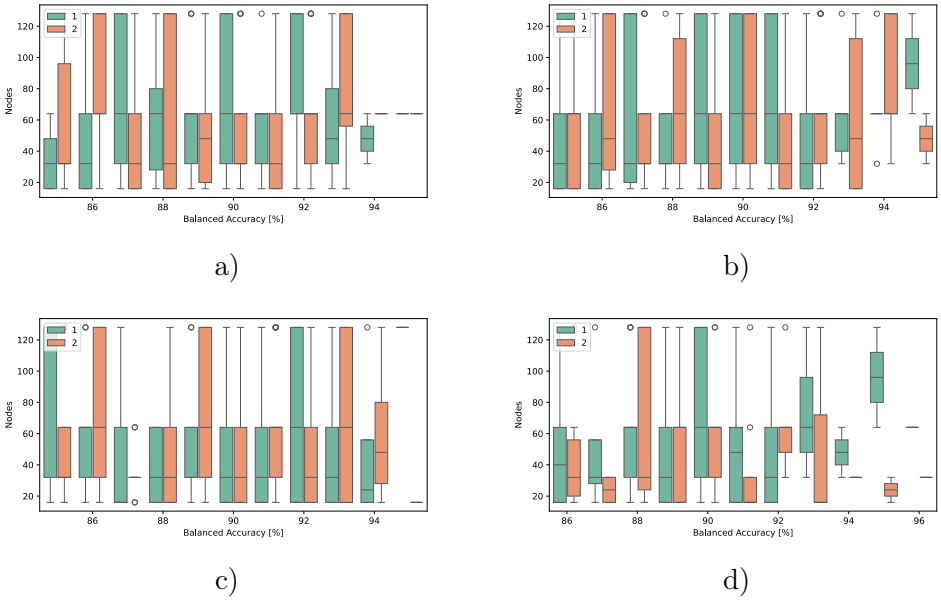

Figure 8: The relationship between number of nodes in the fully connected hidden layers and BAC. For a) $N = 2$, b) $N = 3$, c) $N = 4$ and d) $N = 5$.

Table 1 shows the parameters for the model that performed best for each N. To assess which model is best, the balanced accuracy was calculated for each model. The BAC is listed on the far right of the table.

N	K1	K2	K3	K4	K5	F1	F2	F3	F4	F5	FC1	FC2	Dropout	Max BAC
2	20	50	-	-	-	30	50	-	-	-	64	64	0.6	0.949
3	10	25	50	-	-	50	20	15	-	-	128	32	0.6	0.946
4	30	20	50	20	-	5	25	40	50	-	128	16	0.2	0.946
5	40	40	10	5	20	5	25	40	40	50	64	32	0.3	0.956

Table 1: Parameters for the most optimal models found under the small random search.

In figure 9 the confusion matrices of the four models in the table are shown. All four models have quite good results, both in terms of the balanced accuracy and the confusion matrices. The model with five convolutional layers has the best results, with $BAC = 0.956$ and values on the diagonal of the confusion matrix very close to one.

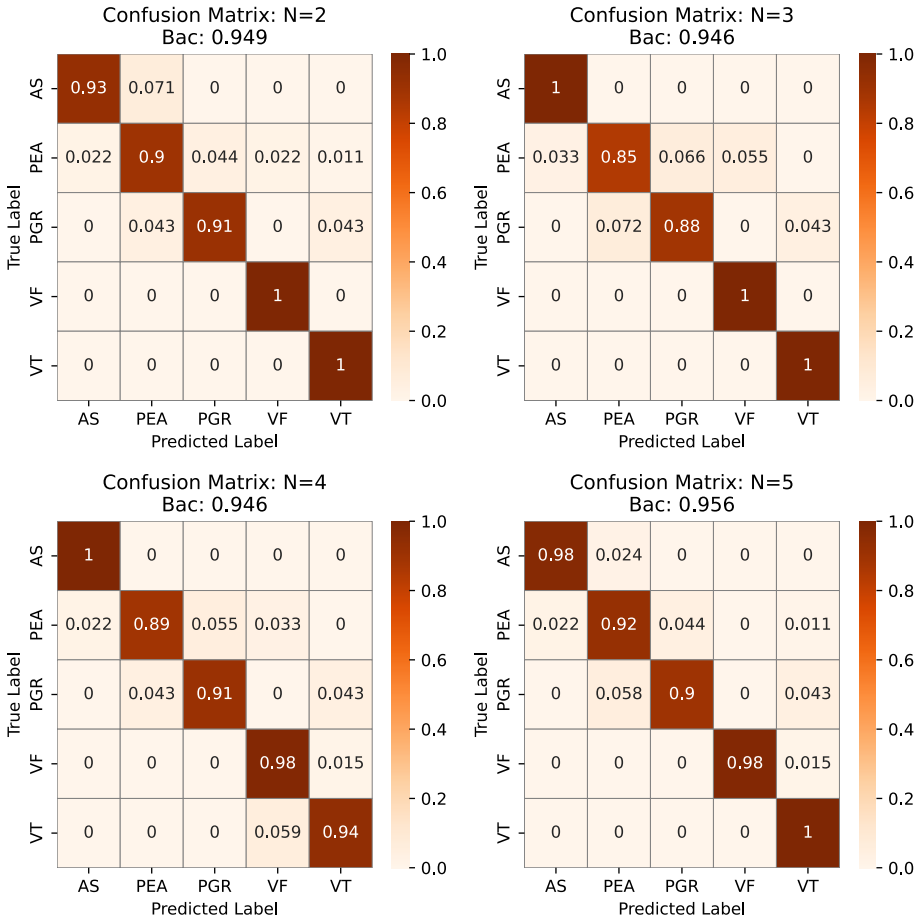

Figure 9: Confusion matrices of the best model for each N.

4.2 With fixed parameters

In this section the dropout and number of nodes in the fully connected layers were fixed, to $Dropout = 0.3$, $FC_1 = 64$ and $FC_2 = 32$. These experiments where done to try to get a better overview of the effect that kernel size and filters have on the model.

Figure 10 shows the relation between the number of tuneable parameters and the accuracy, for each N. The plot on the left uses accuracy and the plot on the right uses balanced accuracy. It is clear that these results are better than the results without fixed parameters, because none of the models get the BAC of 20%. Apart from that one can still observe that more trainable parameters does not necessarily increase the accuracy.

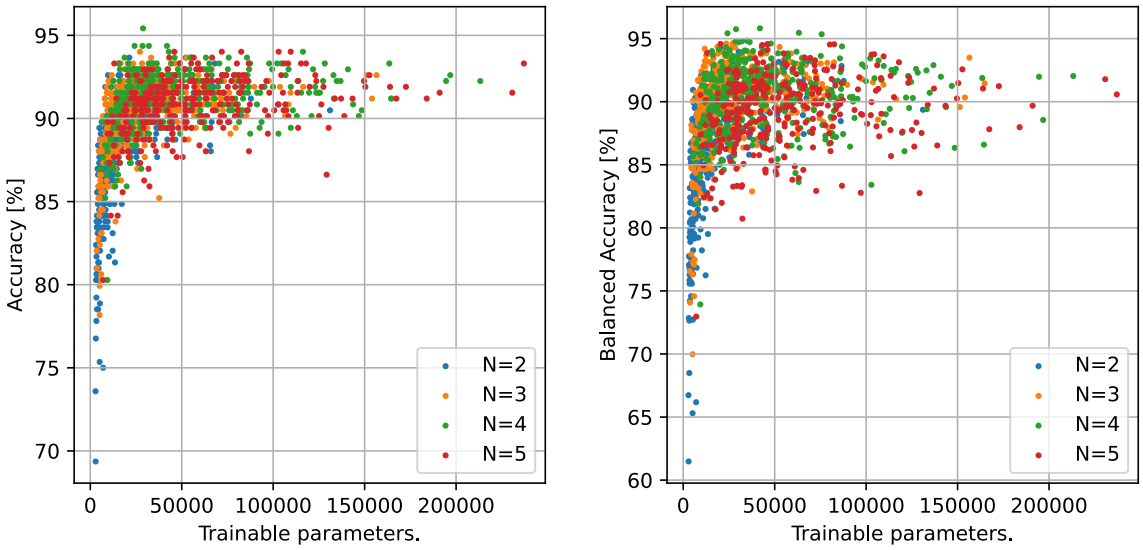

Figure 10: Scatter plots showing the relationship between number of trainable parameters and the accuracy (on the left) or the balanced accuracy (on the right), for each N, with fixed parameters.

In figure 11 the relation between the kernel size and the BAC is shown. It is still hard to see any real correlation between kernel sizes and results for the bigger N's, but for $N = 2$ and $N = 3$ the accuracy gets better for bigger kernel sizes.

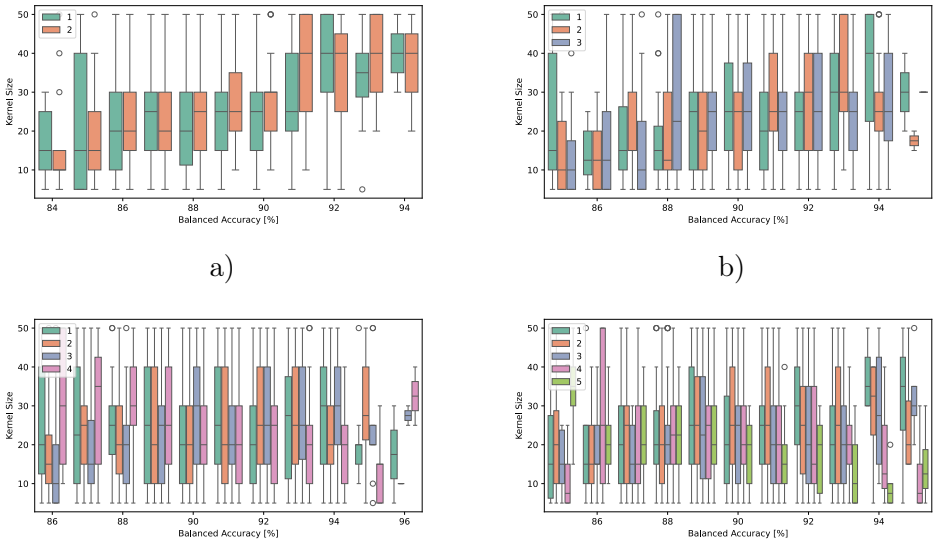

Figure 11: The relationship between kernel size and BAC with fixed parameters. For a) $N = 2$, b) $N = 3$, c) $N = 4$ and d) $N = 5$.

The relation between the number of filters and the BAC are plotted in figure 12. This plot does not show any obvious relation between the number of filters and the BAC, so the best number of filters was determined from which model got the best BAC for each N.

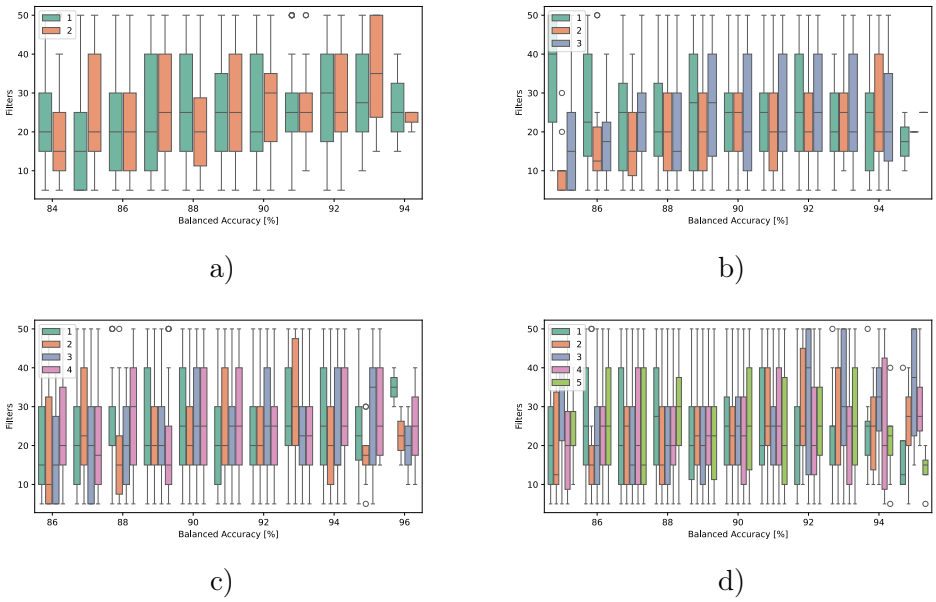

Figure 12: The relationship between number of filters and BAC with fixed parameters. For a) $N = 2$, b) $N = 3$, c) $N = 4$ and d) $N = 5$.

The best models, for each N, found from testing with fixed Dropout and FC_j are listed in table 2. The model with the best BAC is highlighted in green, and this is the one that is used in the next subsection. It got a BAC of 0.958.

N	K1	K2	K3	K4	K5	F1	F2	F3	F4	F5	Max BAC
2	40	20	-	-	-	40	20	-	-	-	0.942
3	25	20	25	-	-	20	15	30	-	-	0.946
4	40	30	10	40	-	30	10	25	40	-	0.958
5	40	5	25	25	20	40	15	30	5	10	0.946

Table 2: Parameters for the optimal models found under random search with fixed parameters: $Dropout = 0.3$, and dense layers $FC_1 = 64$ and $FC_2 = 32$. With two to five convolution blocks in the models. The green highlights the model with the best BAC.

Confusion matrices of the best models for each N are shown in figure 13. This supports the table 2 in that the model with $N = 4$ is the best, as the values on the diagonal are so close to one.

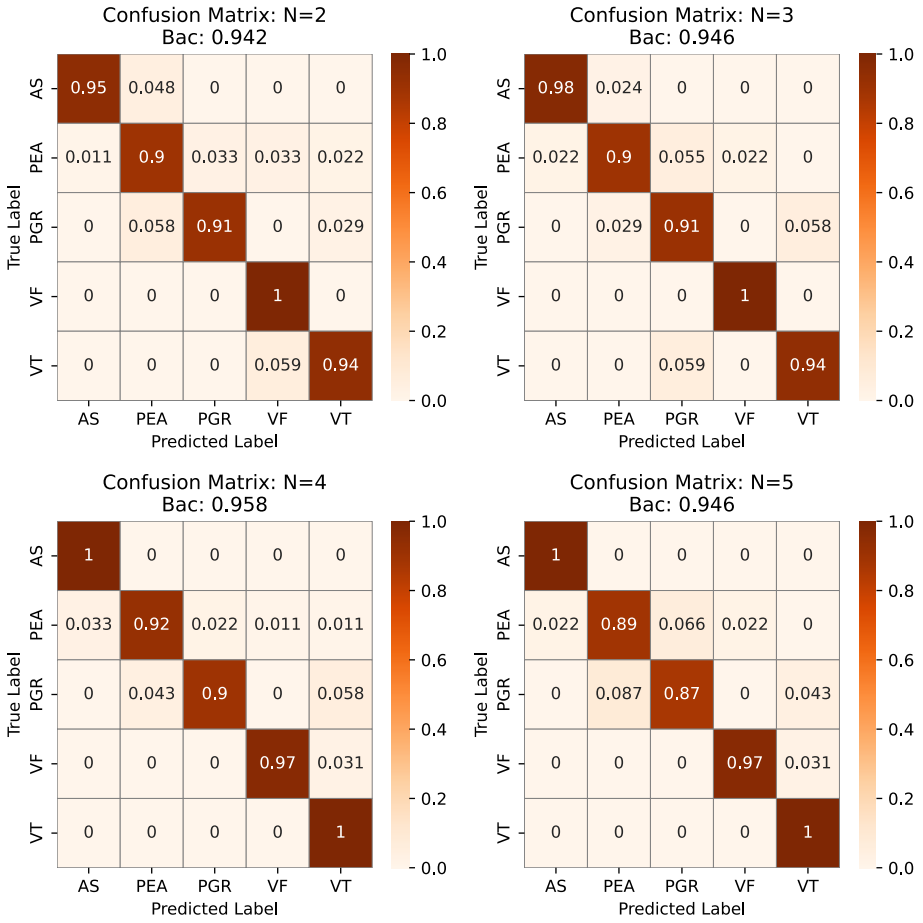

Figure 13: Confusion matrices of the best model for each N, with dropout fixed to 0.3, and the fully connected layers fixed at $FC_1 = 64, FC_2 = 32$.

4.3 Best model

To evaluate the best model as found with the fixed random search (green in table 2), the model was tested again to see how it performs over 10-fold cross validation.

Figure 14 shows all ten training histories when retraining with the best model. From the validation accuracy and the validation loss it is a noticeable difference between each run.

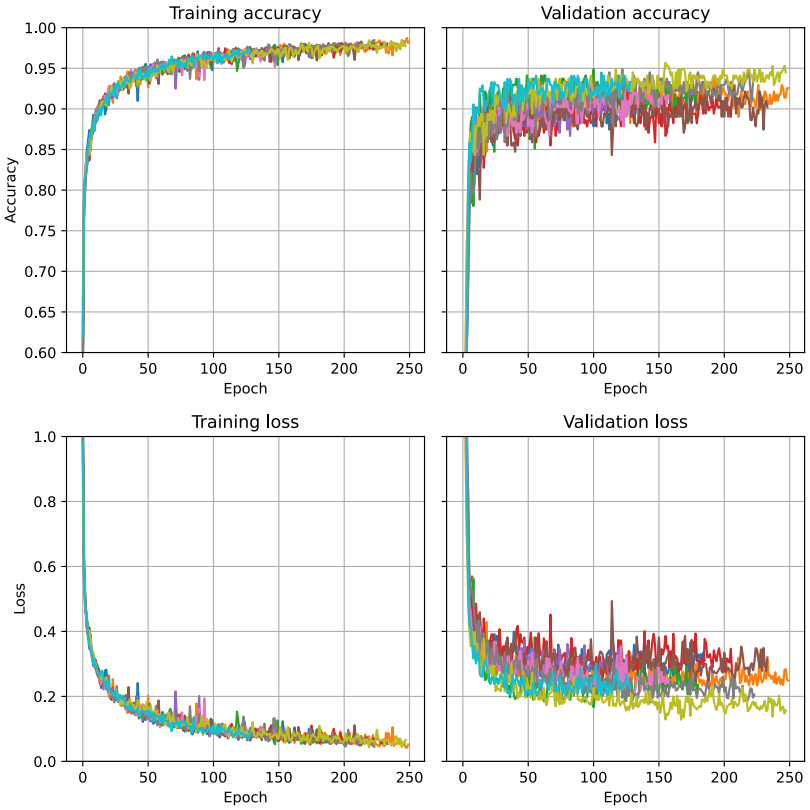

Figure 14: Training histories for all ten runs with the best model.

Table 3 shows the accuracy, BAC, sensitivity (Sen) and positive predictive value (PPV) for the best model, with median and 25th/75th percentile. The sensitivity is the number of true positives divided by the summation of true positives and false negatives, i.e. it shows if the model can find all objects in the class. The PPV is the number of true positives divided by the summation of true positives and false positives, i.e. it shows how often the model is correct when it predicts this class. The values for ventricular tachycardia (VT) are quite low for both the sensitivity and PPV. This could be because it has a very low number of samples in the training set. The BAC reaches a value of 0.91, which is a bit lower than in the previous subsection. This can indicate that the model has a lot of variance due to randomness in data selection and weight initialization in the model.

	Acc	BAC	AS		PEA		PGR		VF		VT	
			Sen	PPV	Sen	PPV	Sen	PPV	Sen	PPV	Sen	PPV
Median	0.92	0.91	1.00	0.95	0.91	0.93	0.87	0.92	0.98	0.95	0.82	0.74
25th percentile	0.91	0.91	0.98	0.95	0.90	0.92	0.86	0.92	0.97	0.93	0.78	0.69
75th percentile	0.93	0.92	1.00	0.95	0.92	0.94	0.88	0.94	0.98	0.96	0.82	0.76

Table 3: Table containing the median and the 25th/75th percentile of accuracy, BAC, sensitivity (Sen) and positive predictive value (PPV) of every class when retesting the "best model".

5 Discussion

In general this type of network seems to work well for classifying heart rhythms, with many of the models reaching over 90% BAC. This section will discuss the outcome of the project and some areas where there is potential for improvement.

To achieve more robust and better results, a larger dataset could be desirable. There does not exist a specific rule that says how big a dataset should be for a CNN, but most sources agree that it should be as big as possible. Also, as mentioned in section 3, the balance of the dataset could be better. Especially the VT samples are very few, and this leads to the model having a harder time classifying this class correctly.

With the current model architecture it was not possible to have more than five convolution blocks. This is because most trials would fail with more convolution blocks due to the down-sampling. A possible improvement is to have smaller kernels in the later layers, and/or use padding with the max pooling layers. This would make it possible to have more convolution blocks.

Future work could also experiment more with the fully connected layers. This was not experimented much with, and as [1] says, bigger fully connected layers probably would improve the model. There could also be made a few improvements to how the parameters are tested. Often the size of both filter and kernels gets smaller for each layer, while in this model the size was chosen randomly.

5.1 Conclusion

This project has developed a CNN model for classifying heart rhythms. It has been tested, first with five non-fixed parameters and then with three. The best model was achieved with four convolutional blocks, which gave a balanced accuracy of 0.91. The biggest improvements to the model from the previous project [9], is the added global max pooling after the convolutional blocks, and bigger kernel sizes in the convolutional layer. This resulted in a more robust model with less overfitting and better test results.

The same dataset is used in the articles "A Convolutional Neural Network Approach for Interpreting Cardiac Rhythms from Resuscitation of Cardiac Arrest Patients" [7] and "Multimodal Biosignal Analysis Algorithm for the Classification of Cardiac Rhythms During Resuscitation" [11]. Compared to the results from these articles, this project has achieved improvements in Acc, Sen and PPV. This can especially be seen in the Sen and PPV of the VT class. Where in the first article Sen = 0.60 and PPV = 0.481, the second Sen = 0.773 and PPV = 0.661, and in this project Sen = 0.82 and PPV = 0.74. This shows that the project have made progress in the study of classifying heart rhythms.

References

[1] N. C. . affiliates. Linear/fully-connected layers user's guide. URL [Online; accessed November-2023].

[2] S. Allwright. Accuracy vs balanced accuracy, which is the best metric? URL [Online; accessed November-2023].

[3] E. Burns and R. Buttner. Normal sinus rhythm, 2021. URL [Online; accessed November-2023].

[4] C. Clinic. Pulseless electrical activity, . URL [Online; accessed November-2023].

[5] C. Clinic. Asystole, . URL [Online; accessed November-2023].

[6] M. Clinic. Ventricular tachycardia, . URL [Online; accessed November-2023].

[7] T. Eftestøl, M. A. Hognestad, S. A. Søndeland, A. B. Rad, E. Aramendi, L. Wik, and J. Kramer-Johansen. A convolutional neural network approach for interpreting cardiac rhythms from resuscitation of cardiac arrest patients. *Computing in Cardiology, vol. 50*, 2023.

[8] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[9] M. A. Hognestad and S. A. Søndeland. Cardiac rhythm interpretation. *Universitetet i Stavanger*, 2022. URL.

[10] V. Krasteva, S. Ménétré, J.-P. Didon, and I. Jekova. Fully convolutional deep neural networks with optimized hyperparameters for detection of shockable and non-shockable rhythms. *Sensors*, 20, 2020. URL [Online; accessed November-2023].

[11] H. Lasa, U. Irusta, T. Eftestol, E. Aramendi, A. B. Rad, J. Kramer-Johansen, and L. Wik. Multimodal biosignal analysis algorithm for the classification of cardiac rhythms during resuscitation. *Computing in Cardiology, vol. 47*, 2020.

[12] M. L. Mastery. A gentle introduction to the rectified linear unit (relu). URL [Online; accessed November-2023].

[13] J. H. Medicine. Ventricular fibrillation. URL [Online; accessed November-2023].

[14] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi, et al. Kerastuner, 2019. URL [Online; accessed November-2023].

[15] N. T. Srinivasan and R. Schilling. Sudden cardiac death and arrhythmias. *AER*, 2018.

[16] O. Universitetssykehus. Hjertestans, 2020. URL [Online; accessed November-2023].

[17] L. Wik, J. Kramer-Johansen, H. Myklebust, H. Sørebø, L. Svensson, B. Fellows, and P. A. Steen. Quality of cardiopulmonary resuscitation during out-of-hospital cardiac arrest. *JAMA*, 293, 2005.