

# A Study of ECG Heartbeat Categorization

Minh Pham Quang - 23BI14296

## I. INTRODUCTION

This report presents the results of my study on ECG heartbeat categorization taken from Kaggle. The data used in this study is the MIT-BIH Arrhythmia Database. I tried to analyze on the data and apply 3 models based on Random Forest, CNN to classify 5 types of heartbeats.

## II. DATA ANALYSIS

The data used in this study is the MIT-BIH Arrhythmia Database. The data consists of 5 types of heartbeats: Normal, Supraventricular, Ventricular, Fusion and Unknown, which are encoded to 0, 1, 2, 3 and 4 respectively. Each data contains 187 values of the heartbeat signal.

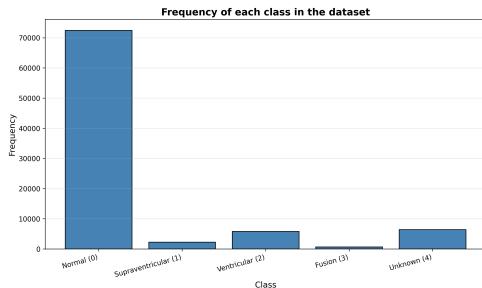


Fig. 1: Frequency of each class in the dataset

As we can see from the histogram at figure 1, the training dataset in Kaggle is extremely bias to the Normal class, which is the most common type of heartbeat. This biasness can lead to the model being overfitted to the Normal class.

To avoid overfitting, the up-sample technique to balance the dataset was used. When divided the total number of files in the training dataset by 5, I obtained about 17510. So I balanced the dataset by resampling each class to have around 17500 samples in each class.

Besides, I also split the test dataset in Kaggle into 2 parts: 50% for validation and 50% for testing. The distributions of 2 these datasets can be seen in the figure 3.

## III. MODEL

For categorization tasks, I tried to implement 5 different classification models which take the 187 values of the heartbeat signal as input and output the category of the heartbeat.

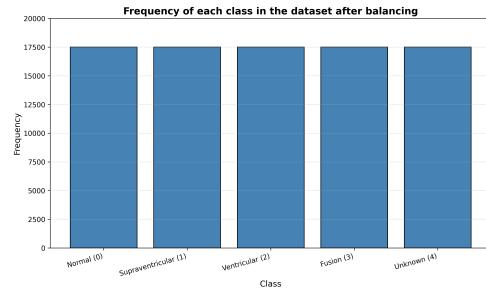


Fig. 2: Frequency of each class in the dataset after balancing

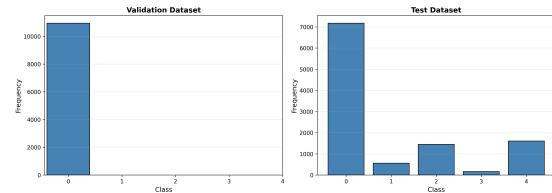


Fig. 3: Frequency of each class in the valid and test dataset

### A. Random Forest

I began with a Random Forest classifier due to its computational efficiency and effectiveness with structured data. This model was trained with 100 decision trees and a random seed of 42 for reproducibility, without utilizing the validation set during training.

### B. Convolutional Neural Network

I tried to implement a simple convolutional neural network due to the powerful of the concept of convolution operation combined deep learning model which seem to be useful for signal processing. The network contains three 1D convolutional layers for feature extraction of the signal and two fully connected layers.

### C. Convolutional Neural Network with Residual Connection

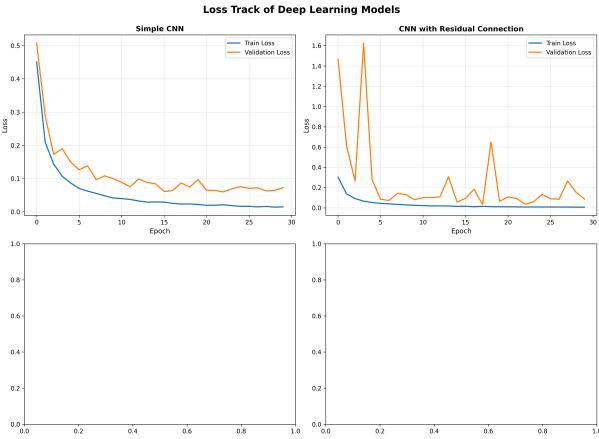
The third model is a convolutional neural network with 1 first Conv1D layer and 5 residual blocks. The reason I used this model is that the skip connection can help it to prevent the vanishing gradient problem, which avoids the model from being overfitted. Besides, I also add some special layers such as BatchNormalization and Dropout to improve the model's performance.

## IV. EVALUATION AND RESULTS

### A. Loss Track of Deep Learning Models

As can be seen from the plots of loss for all models, the train loss of all models are decreased nearly to 0. However, the CNN with Residual Connection seems to tackle the overfitting problem. The reasons for that is easily understand by the mechanism of Residual Block which is avoiding the vanishing gradient problem.

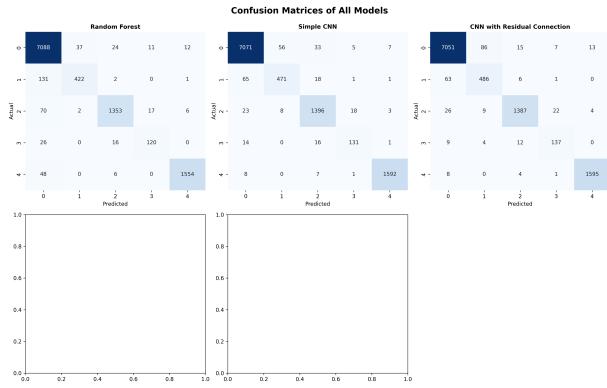
TABLE I: Loss track of the Deep Learning models



### B. Confusion Matrices of all Models

For the results recorded in the testing dataset, CNN with Residual Connection also shows the best prediction for unseen data.

TABLE II: Confusion Matrices of all Models



## V. COMPARISON WITH STATE-OF-THE-ART

We benchmark our models against Kachuee et al. [1], a widely cited study on ECG heartbeat classification using the MIT-BIH Arrhythmia Database.

### A. Overview of Kachuee et al.'s Method

Kachuee et al. proposed a deep CNN with 5 residual blocks, batch normalization, pooling, and data augmentation. Their model includes:

- Initial Conv1D layer (kernel size 7)
- Five residual blocks with skip connections
- Two fully-connected layers (32 neurons each)
- Softmax output for 5-class classification (N, S, V, F, Q)

They also explored transfer learning for myocardial infarction detection.

### B. Performance Comparison

Table III compares our results with Kachuee et al.

TABLE III: Classification accuracies compared with Kachuee et al. [1]

| Model                 | Accuracy (%) |
|-----------------------|--------------|
| Kachuee et al. (2018) | 93.4         |
| Our Random Forest     | 96.26        |
| Our Simple CNN        | 97.40        |
| Our ResNet CNN        | 97.35        |

### C. Key Insights

- Architecture:** Our ResNet CNN is inspired by Kachuee et al. but varies in depth and configuration.
- Training:** We train from scratch, while Kachuee et al. also apply transfer learning.
- Performance:** Residual connections consistently improve accuracy; deep learning outperforms Random Forest baselines.
- Class Imbalance:** Both studies address imbalance via augmentation.

## VI. CONCLUSION

This study presented a comprehensive analysis of ECG heartbeat classification using the MIT-BIH Arrhythmia Database. We implemented and compared three different models: Random Forest, Simple CNN, CNN with Residual Connection.

Our results demonstrate that deep learning models, particularly those with residual connections, are effective for ECG heartbeat classification. The CNN with Residual Connection showed the best performance, successfully addressing the overfitting problem through skip connections and regularization techniques.

Comparison with the state-of-the-art work by Kachuee et al. (93.4% accuracy) provides context for our results and validates the importance of residual architectures for this task. Future work will focus on matching or exceeding this benchmark through hyperparameter optimization, transfer learning, and ensemble approaches.

## REFERENCES

- [1] Kachuee, M., Fazeli, S., and Sarrafzadeh, M. (2018). ECG Heartbeat Classification: A Deep Transferable Representation.