Project Report
ACDC Challenge – Supervised Learning

Sacha Khosrowshahi – Télécom Paris

Contents

1 Challenge Overview

The goal of this challenge was to find an optimized and efficient way to classify patients among 5 different classes of diagnosis : Healthy controls, Myocardial infarction, Dilated cardiomyopathy, Hypertrophic cardiomyopathy, Abnormal right ventricle. In order to do so we are given a set of data containing their CMRI and a set of partial segmentations : For each subject, two images are provided at two different time points along the cardiac cycle: one image at end diastole (end of dilation) and one image at end systole (end of contraction). Each MRI image consists in a 3D volume containing the heart and adjacent structures.
It's important to mention that for the testing set the left ventricule segmentation is missing and therefore we will have to retrieve it using a well-chosen method.

1.1 The dataset

The dataset provided contains the data of 150 patients which is remarkably small given the task. Indeed with such a small dataset we will have to use a method adapted for a small number of features and samples. Moreover the dataset as said in the precedent section contains segmentations that we need in order to extract the features recquired. In order to know what we need and how we are going to proceed to get it, let's first discuss the different methods proposed to do the diagnosis and follow along the line of the chosen method.

1.2 Objective

It can be mentioned, without any further action taken, that in such a context of classifying disease we are looking for specific True Positive and False Positive ratios. Indeed, when we want to classify patients among classes of disease it can be rational to consider methods that have a high TPR at the expense of a higher FPR. We would like to detect the highest possible number of disease at the expense of detecting ones that aren't there.

We now understand that the goal of this challenge is to find an optimized way to get the highest TPR possible given a modest dataset and some missing features.

2 Method Selection

2.1 Available Methods

The challenge overview provided several scientific papers. Among them, I selected: **"Automatic Segmentation and Disease Classification Using Cardiac Cine MR Images" (Springer, 2017)**

2.2 Justification

All the methods proposed in the challenge except one used neural networks for the classification parts. Indeed we already have access to most of the segmentations therefore we don't take into account the neural networks there to segment.

Moreover the reason why I chose to go for the automatic method that only used a random forest is mainly because of interpretability. Indeed in the context of disease classification we want to make sure the model we're using can give us analysis on the reason some features are more important than others in order to improve its performances. In the case of a neural network, working like a black box, if a part of a CMRI image triggers the model to do a wrong classification we can't know it and

therefore we may get stuck. The features with this method are easily interpretable and we will be therefore able to do feature engineering.

I chose the Random Forest method because it was also efficient on it's results. Obtaining an overall accuracy of 91 percent is more than enough for a target percentage.

Ultimately, the fact that we don't have a lot of samples in our dataset would give poor result when using a neural network.

3 Data Preprocessing

3.1 Loading and Visualization

MRI volumes and masks are loaded using `nibabel`. 2D slices are extracted for display with `matplotlib`.

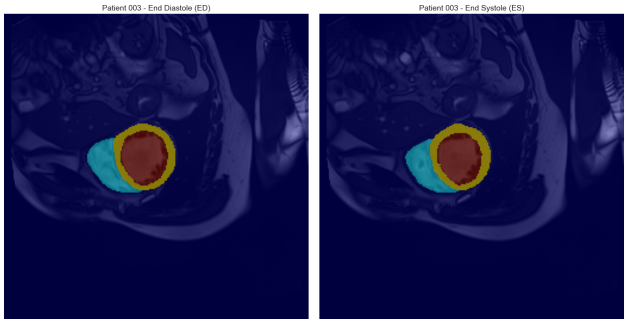
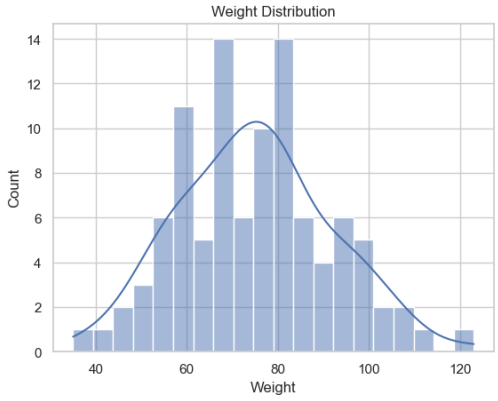
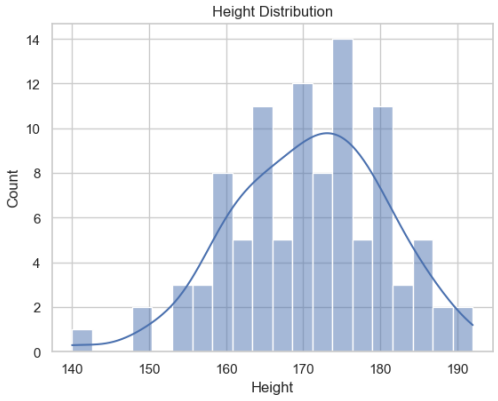


Figure 1: ED-ES Visualization

We also want to see the distribution of heights and weights for the patients :



(a) Weight distribution



(b) Height distribution

Figure 2: Distributions among patients

3.2 Feature Extraction

To build a feature set suitable for supervised learning, we extracted clinically meaningful anatomical measurements from the 3D segmentation masks provided at two time points in the cardiac cycle: end-diastole (ED) and end-systole (ES). These segmentations label three cardiac structures: the right ventricle cavity (label 1), the myocardium (label 2), and the left ventricle cavity (label 3).

However, in some cases, the left ventricle segmentation was missing or incomplete; we addressed this by reconstructing the left ventricle mask slice-by-slice using a morphological hole-filling algorithm applied within the myocardium boundaries.

For each subject, we computed the volume of each anatomical structure at both time points by summing the number of voxels corresponding to each label and multiplying by the voxel volume extracted from the NIfTI header. These volumes were converted from mm^3 to liters. From the ED and ES volumes of each ventricle, we calculated the ejection fraction (EF) as:

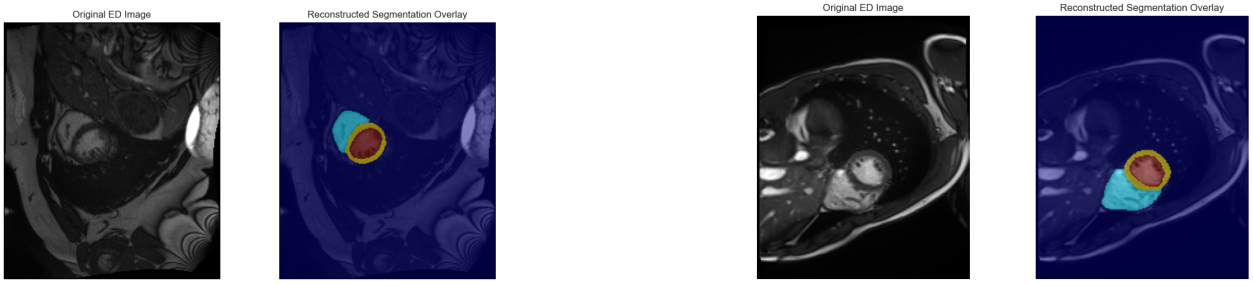
$$EF = \frac{V_{ED} - V_{ES}}{V_{ED}} \times 100$$

where V_{ED} and V_{ES} are the volumes at end-diastole and end-systole respectively. This metric is critical in cardiology for assessing heart function.

We also derived additional ratios to capture structural relationships:

- The ratio of right to left ventricle volume (RV/LV) at both ED and ES.
- The ratio of myocardium to left ventricle volume (Myo/LV) at both ED and ES.

Finally, each patient's height and weight were included from the metadata, enabling future computation of body surface area or indexing features by body size if necessary. All extracted features were compiled into a structured DataFrame used as input for model training.
We made sure the reconstruction worked by vizualising it after and comparing it to ground-truths patients :



(a) Random patient 131 reconstructed

(b) Random patient 146 reconstructed

Figure 3: Reconstructions of the LV

4 Model Training and Evaluation

To perform the classification task, we selected a Random Forest classifier due to its robustness, interpretability, and ability to handle non-linear relationships without requiring extensive preprocessing. We configured the model to balance class weights to mitigate the effects of class imbalance in the dataset.

We then performed a hyperparameter search using `RandomizedSearchCV`, testing 50 different combinations of key parameters, including the number of estimators, maximum tree depth, minimum samples per split, and leaf size. A 5-fold stratified cross-validation scheme was used to ensure that each fold preserved the distribution of the diagnostic classes. The best-performing model achieved a cross-validated accuracy of `95%` and the optimal hyperparameters were retained for final evaluation.

To assess the classifier's ability to generalize across all classes, we conducted another 5-fold stratified cross-validation using the selected model. For each fold, we computed the precision, recall,

and F1-score per class using the `classification_report` function from Scikit-learn. These metrics were averaged across folds, enabling us to identify class-wise performance patterns and potential weaknesses (e.g., systematic confusion between similar cardiomyopathies).

The results were visualized as bar plots for each metric across all five diagnostic categories. These plots provide a clear overview of the model's performance and highlight classes that may require further refinement through feature engineering or data augmentation.

Finally, we computed the global cross-validation accuracy across the entire training set, yielding a mean accuracy of `95%`. The best estimator was then retrained on the full training dataset to prepare for test-time inference and leaderboard submission.

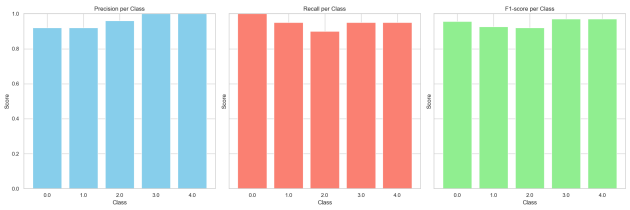


Figure 4: Score per class

Seeing no improvements in the first tries here are some of the changes I made through the process : I first did the reconstruction also on the training set because I thought there could be a bias to only have the reconstruction on the testing set which could make sense in case the reconstruction isn't perfect.

Moreover I did some feature engineering in order to enrich the dataset by adding some features that took into account body size ratios.

Finally I made some cross-validation and refining to see how well the model was performing on each class in order to see if I could detect a physical aspect of a certain disease that would entail the model to fail.

5 Conclusion

In this project, we developed a supervised learning pipeline for automated classification of cardiac pathologies using cine-MRI data. By leveraging interpretable anatomical features—such as ventricular volumes, myocardial thickness, and ejection fractions—we trained a Random Forest classifier capable of achieving high accuracy across five diagnostic classes. Our approach prioritized clinical relevance, interpretability, and robustness, making it well-suited to the limited size of the dataset. Through cross-validation and performance analysis, we demonstrated the model's generalization ability and identified class-level strengths and weaknesses. Future work could explore integrating automated segmentation models or dynamic features extracted from the full cardiac cycle to further improve diagnostic precision.