IM05 - Kaggle Challenge Report
Cardiac Pathology Prediction

Aymane Hamdaoui
Télécom Paris

May 2025

Abstract

This report presents our solution to the Kaggle Challenge on Cardiac Pathology Prediction, which aims to classify patient health status (healthy/ill), and identify the most probable pathology. Our approach synthesizes current literature through a methodological interpolation of existing techniques, followed by qualitative analysis of the implemented solution.

1 Introduction

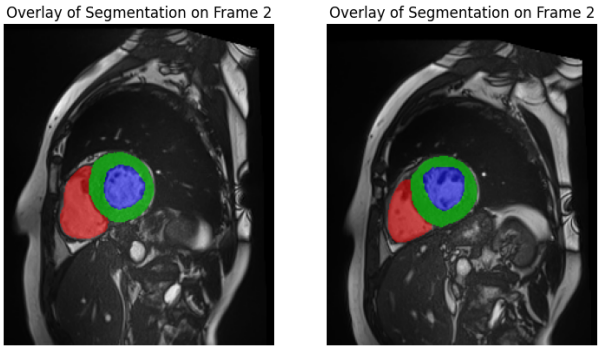
The *Cardiac Pathology Prediction* aims to predict patient's cardiac pathologies. This task is critically important in cardiac pathologies diagnosis as evidenced by the extensive body of literature dedicated to it as we will see in the state of the art in the following.

In this report, we:

- Propose an implementation of our method.
- Analyze the dataset and highlight key preprocessing steps and key features.
- Explain how we achieved a high accuracy.

2 Dataset description

The data used consists of 150 MRIs and their corresponding segmentations of the Myocardium (MYO), the Left Ventricle (LV), and the Right Ventricle (RV) 1. We are given two files per patient, one .nii file at end diastole and one at end systole, which represent the maximum and minimum contraction for the LV. Moreover, we have access to the height and the weight of each patient. For the first 100 patients we have access to the target, these are the training data. For the 50 other ones, the target and the segmentation of the LV are missing. They make up the testing data. The metric chosen in order to score the performances of our models is accuracy. When predictions are submitted, a public score is computed on approximately 30% of the test data and immediately disclosed. The remaining 70% constitutes a private test set, whose scores remain undisclosed until the challenge concludes to prevent potential reverse engineering of the evaluation metric.



((a)) Slice 2 at ED ((b)) slice 3 at ED

Figure 1: Ground Truth (i.e. given segmentation) overlapped to the initial MRI (Patient ID: 027). LV in blue, MYO in green and RV in red.

Both the training and the testing data are balanced, i.e. each class is represented equally. There are 5 categories that could be predicted:

- '0' - Healthy controls
- '1' - Myocardial infarction
- '2' - Dilated cardiomyopathy
- '3' - Hypertrophic cardiomyopathy
- '4' - Abnormal right ventricle

The frequency of occurrence of each class is 20%.

From now on the goal is to deal with the following problems: how to get the LV segmentation on our testing dataset? How to extract good features? How to deal with this small amount of data to make sure our model is good and does not overfit? And which model should we use?

Some of these questions are already addressed in the state of the art.

3 State of the art

In order to address this challenge, 4 papers were mentioned to us to start with. But our researches led us to other interesting papers.

3.1 Non-exhaustive overview of the literature

- **Wolterink et al.** used 14 features: LV, RV, and myocardial volume at ED and ES (in mL), the LV and RV ejection fraction (EF), the ratio between RV and LV volume at ED and ES, and the ratio between myocardial and LV volume at ED and ES [Wol+18]. Then a Random Forest classifier model was trained on these features.

- **Isensee et al.** extracted a series of instants and dynamic features from the segmentation maps [Ise+18] . They also used a Random Forest classifier.

- **Khened et al. (2018)** used 11 features: Ejection Fraction of LV and RV, Volume of RV and LV at ED and ES, Mass of MYO at ED and Volume at ES, Height, and Weight [KAK18]. They also used a RF classifier.

- **Khened et al. (2019)** extended their previous work by developing a two-stage classification approach. The first step is an ensemble hard voting classifier made of an SVM, a Random Forest, a Naive Bayes and an MLP. The features used at this stage are similar to the previous ones. Then an MLP which plays the role of the expert classifier is used to refine labels 1 and 2. [KKK19].

3.2 Comparative Review

Bernard et al. conducted a comprehensive review of these papers except the last one, classifying the ACDC challenge submissions and comparing their methodologies [Ber+18].

From Figure **??**, one can observe that [KAK18] achieved near-perfect classification with 48 out of 50 patients correctly identified. Figure **??** displays the best results on this classification challenge. Notably, the second and third best methods closely followed this performance with 92% accuracy each. But [KKK19] claims achieving a perfect accuracy score on classification.

Methods		***Accuracy***
Authors	Architectures	
Khened *et al.* [46]	Random Forest	**0.96**
Cetin *et al.* [53]	SVM	0.92
Isensee *et al.* [44]	Random Forest	0.92
Wolterink *et al.* [50]	Random Forest	0.86

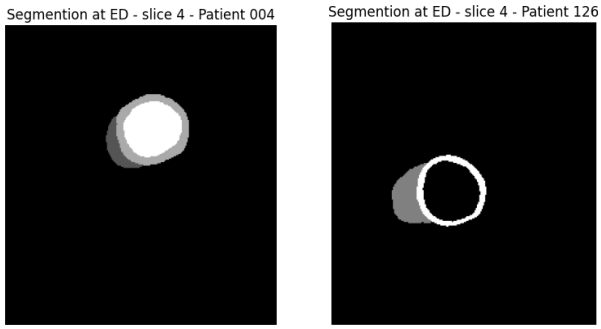
Figure 2: Performance comparison on the Classification Challenge from [Ber+18]

Therefore, according to [Ber+18] **Khened et al.** the best results. However, this paper does not mention [KKK19] which claims to achieve a perfect score of 100% on the 50 patients from the test. This new method improves the result of their previous implementation (i.e. [KAK18]).

We will try implementing some of these methods in order to compare them.

4 Data Preprocessing

Raw data is given to us as explained in our **Data description**. The first issue we have to deal with is the LV missing in the testing data set. Figure 3 shows this issue.



((a)) Segmentation training data (Patient ID: 004)

((b)) Segmentation on testing data (Patient ID: 126)

Figure 3: Comparative segmentation showing (a) training and (b) testing.

This problem is addressed with the following pipeline 4. We first binarize the ground truth and find the contours of the mask using *OpenCV*. Then we fill the hole and the difference between the first mask and the filled one gives us our LV.

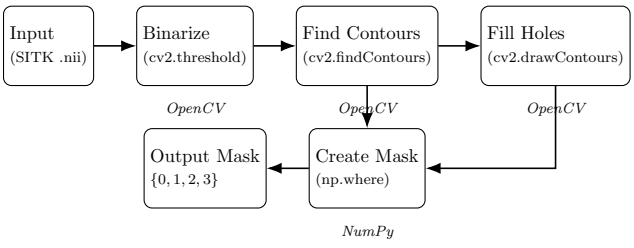


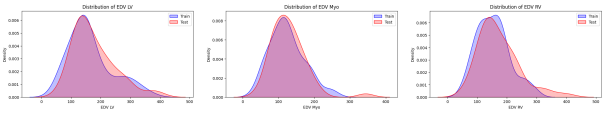
Figure 4: Processing pipeline using SimpleITK (SITK) OpenCV (cv2) and NumPy (np) to get the LV segmentation on testing patient.

Once we have our fully segmented data, we have to extract features to train our models. Different features have been mentioned in the articles mentioned before but a lot of them are redundant. Let's overview them.

We analyze the distribution of key cardiac features across the training and test sets to assess potential discrepancies that could influence model generalization.

4.1 End-Diastolic Volume (EDV)

- **EDV LV:** Right-skewed; test set has heavier tail beyond 200 mL.
- **EDV Myo:** Near-Gaussian in both sets; test has lower variance.
- **EDV RV:** Similar shape; test is more skewed with broader tail.

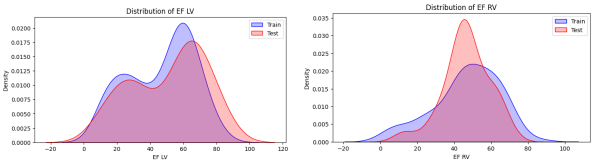


((a)) EDV LV ((b)) EDV Myo ((c)) EDV RV

Figure 5: End-Diastolic Volume distributions

4.2 Ejection Fraction (EF)

- **EF LV:** Bimodal with shift; train set has strong secondary peak.
- **EF RV:** Test set peaks near 50%; train is more spread.



((a)) EF LV ((b)) EF RV

Figure 6: Ejection Fraction distributions

4.3 End-Systolic Volume (ESV)

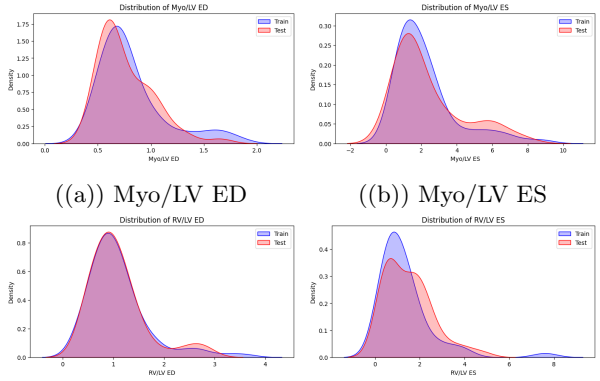
- **ESV LV:** Right-skewed in both; test has more mass in 150–250 mL.
- **ESV Myo:** Nearly identical in both sets.
- **ESV RV:** Test set has a broader high-end tail.

((a)) ESV LV ((b)) ESV Myo ((c)) ESV RV

Figure 7: End-Systolic Volume distributions

4.4 Anatomical Ratios

- **Myo/LV ED:** Peaks just above 0.5; train has broader spread.
- **Myo/LV ES:** Test shows heavier right tail beyond 8.
- **RV/LV ED:** Nearly identical distributions.
- **RV/LV ES:** Test set shows heavier tail up to 9.



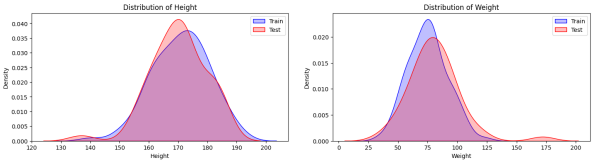
((a)) Myo/LV ED ((b)) Myo/LV ES

((c)) RV/LV ED ((d)) RV/LV ES

Figure 8: Ratios of anatomical regions

4.5 Anthropometric Features

- **Height:** Bell-shaped; test set slightly shifted left.
- **Weight:** Unimodal; test set has heavier right tail.



((a)) Height ((b)) Weight

Figure 9: Distribution of anthropometric features

4.6 Conclusion on features' distributions

Overall, the distributions indicate a relatively balanced train / test split, although subtle differences in the volumes related to EF and RV suggest the need for careful validation. Nevertheless, these distributions show that the extracted features are well-structured and likely relevant, as indicated by their non-random, clinically plausible shapes. This suggests that our model has the potential to generalize well to unseen data, assuming that it is trained with proper regularization and validation strategies. At the same time, the observed shifts and differences of shape in distributions of some features like EF for the RV imply that some variables may not generalize as good as we could think. However, given our small data set, excluding these features might be counterproductive. Moreover, even though shapes could be slightly different, train and test distributions always overlap a lot.

5 Implementation

Having preprocessed the data and validated the consistency of the extracted features across training and

test sets, we now move to the implementation of several classification approaches. Our methodology progresses from simple baseline models to more advanced, literature-inspired architectures, aiming to benchmark their performance.

5.1 First implementation 12 features

My first attempt at this was to extract 12 features and train a model on it. These features were inspired by [Ise+18]. At this time, we did not see MLPs in classes, so we preferred to use classifiers like Random Forests or XGboosts. Training an XGB on these data achieved a mediocre score of 46% on the public. We then tried a classic Random Forest with n=1000 estimatores which was showing some high scores as shown in 10. But it seems that the model was overfitting. The public score was 80% and the private disclosed later was 71% which is really low compared to the state of the art.

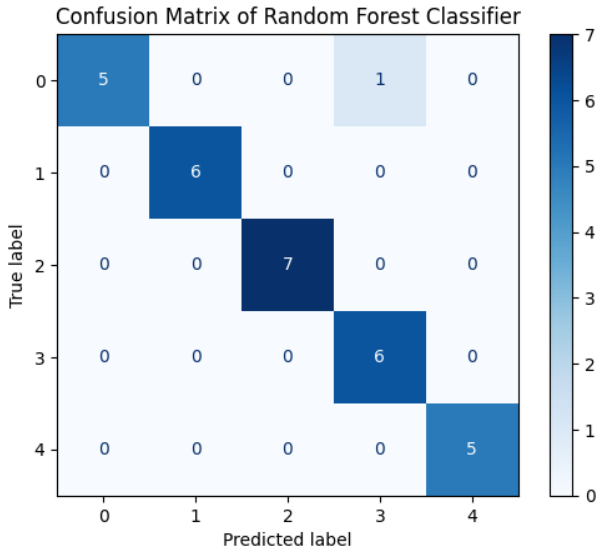


Figure 10: Confusion matrix obtained by training a Random Forest with 1000 estimators on 70% of training data. (Implementation using 12 features)

5.2 Implementation using 14 features

For the second attempt, we tried using more features. Therefore, we extracted the 14 features mentioned in [Wol+18] and trained a Random Forest classifier on this data. In order to choose the parameters we performed a cross validation on our traning data and noticed that even though [Wol+18] uses 1000 estimators, our best scores were obtained using 100 estimators. With this method we obtained an estimator that reached a perfect score on public. From here, it was difficult to get the results any better since we could not see any improvement. Note that we tried other classifiers as Logistic regression, SVM or Naive Bayes but these models were less performant 1. We still wanted to try to find the best parameters for our Random Forest classifier. Therefore, we performed a Grid search in order to maximize accuracy on training set. We obtained a mean accuracy of 96% on our training data but it did not help with the test since the score went down to 86% on public. It means that we were overfitting our data.

Note that we tried different pre-processing tools as scaling or PCA. But RF is invariant to scale changes and PCA could not help. However, these preprocessings helped to increase a bit the performances of other models as GaussianNB which reached 90% percent after a PCA and a Standard Scaling.

Classifier	**Mean Accuracy**	**Std**
Logistic Regression	0.8700	0.0578
SVC	0.9100	0.0510
Random Forest	0.9400	0.0270
Gradient Boosting	0.9400	0.0543
KNeighbors	0.8600	0.0602
MLP	0.8600	0.0632
GaussianNB	0.8600	0.0648

Table 1: Results for each classifier on our training data using a Stratified K fold with 5 splits.

Finally, when the private score was disclosed we saw that this model was achieving an accuracy of 88% on the private. This drop of performance can be explained by the errors of misclassification between class 1 and 2 mentioned by [KKK19].

5.3 Implementation of the 2 stage method

In this part we tried implementing a 2 stage classification as mentioned by [KKK19]. The method was more complex than what we have done until now but implementation was not that difficult. Firstly we had to extract 20 features.

5.3.1 Stage 1 (Voting Classifier)

We use all 20 of the following features as input to each Stage 1 classifier (SVM, RF, GNB, MLP):

1. Volumes at end-diastole (ED) - same as before.
2. Volumes at end-systole (ES) - same as before.
3. Ejection fractions - same as before.
4. Volume ratios at ED and ES - same as before.
5. Myocardial wall-thickness (MWT) variation at ED - a new feature that can be useful in particular because it can describe patient with Myocardial Infarctus.
6. Myocardial wall-thickness (MWT) variation at ES - a new feature that can be useful in particular because it can describe patient with Myocardial Infarctus.

The implementation of this stage was not convincing at first. [KKK19] mentions that only classifiers achieving more than 95% of accuracy were kept but, as we can see in 2, only one classifier achieved such a performance even after optimization with Grid and Bayesian searches. We still used these four classifiers even though they did not match the perfect description of the paper. However we achieved a good performance as shown in 11. We reached an accuracy of 96.7% but it is not extremely relevant since it is obtained on only a few data. We still notice a misclassification between classes 1 and 2 which is addressed in the paper. This is why we might need refinement.

Classifier	Mean Accuracy	Std
Random Forest	0.9400	0.0270
SVC	0.9100	0.0400
MLP	0.9100	0.0274
GaussianNB	0.9400	0.0490

Table 2: Best results for each classifier on our training data with new features using a Stratified K fold with 5 splits.

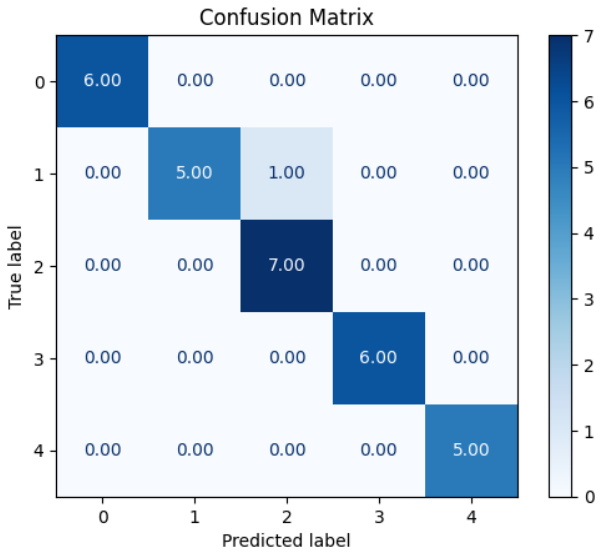


Figure 11: Confusion Matrix obtained after training our Voting Classifier with 70% of the training data.

5.3.2 Stage 2 (MINF vs DCM expert)

Only the end-systolic myocardial wall-thickness variation features are used to refine MINF vs DCM:

1. Maximum of the mean wall thickness across all slices at end-systole.
2. Standard deviation of the mean wall thickness across all slices at end-systole.
3. Mean of the per-slice wall-thickness standard deviations at end-systole.
4. Standard deviation of the per-slice wall-thickness standard deviations at end-systole.

We trained an MLP with two hidden layers each with 100 neurons as described by the paper. After trying to adjust the parameters to avoid overfitting, we obtained a decent score, which is 87% in our validating data. However, we noticed that the MLP did not correctly classify our data. It even got worse, as we can see in figure 12.

The results obtained are far from the ones expected. There might be errors in our implementation, but we could not find them. Still, it is most likely due to the 4 new features that we will discuss in the following section. We still submitted a prediction with this model and obtained a private score of 77% which shows that this model underperformed significantly.

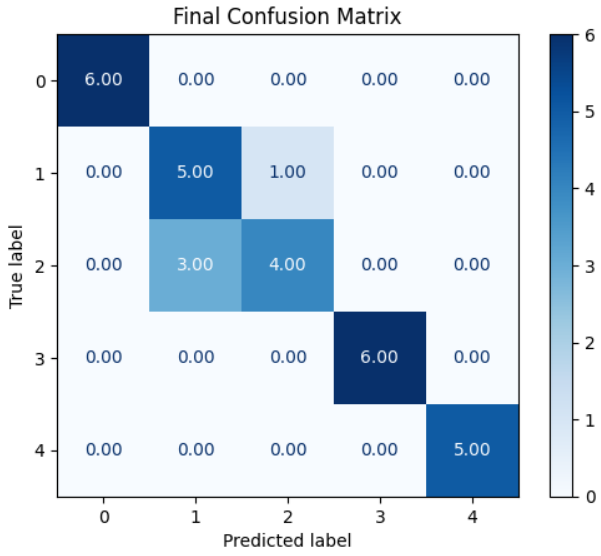
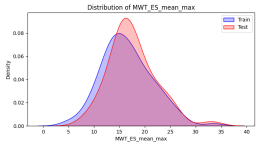


Figure 12: Confusion Matrix obtained after training our 2 stages model with 70% of the training data.

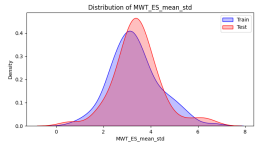
5.4 New features and improvement

5.4.1 New features

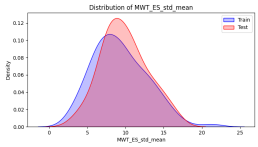
Firstly, after seeing the bad results obtained in the last section, we suspected an issue in the extraction process of the four new features described by [KKK19]. Let's plot them in order to see it if there is a problem:



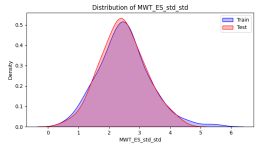
((a)) MWT_ES_mean_max



((b)) MWT_ES_mean_std



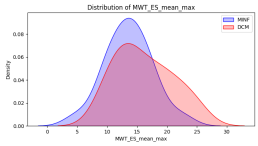
((c)) MWT_ES_std_mean



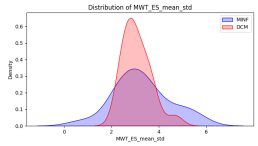
((d)) MWT_ES_std_std

Figure 13: Distributions of MWT-derived features. train in blue vs test in red

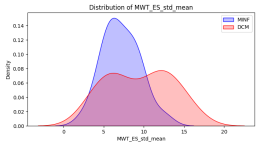
We can notice that distributions from train and testing data are almost the same. This means that the difference between our test score and the private score achieved on the challenge data are most likely due to overfitting. Still, these features could be relevant, at least given these plots. However, they are used mainly to refine our model prediction on classes 1 and 2. Therefore, we have to verify that these features have a high discriminative power between classes 1 and 2.



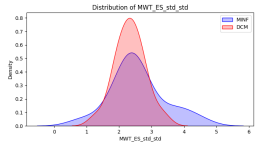
((a)) MWT_ES_mean_max



((b)) MWT_ES_mean_std



((c)) MWT_ES_std_mean



((d)) MWT_ES_std_std

Figure 14: Distributions of MWT-derived features. class 1 in blue vs class 2 in red

Given these new plots, we see that some of MWT derived features are more discriminatuve than others. However, we thought it could be better. In particular, even though they have different heights, their mean/ standard deviations are often close, which makes them overlap a lot.

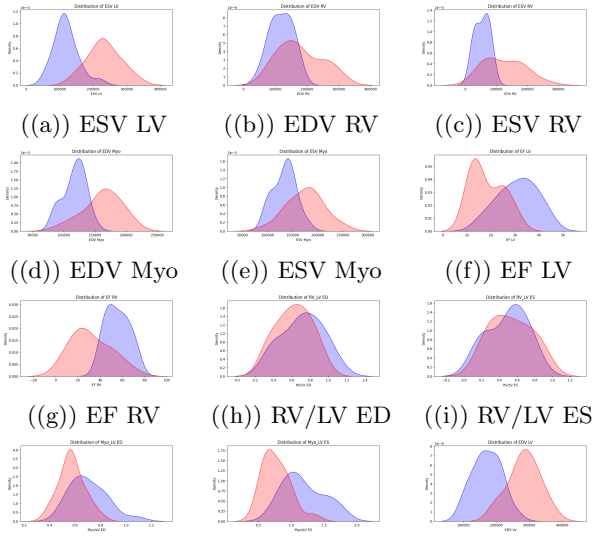
Note that the non-relevance of these features is most likely due to a wrong extraction and not an error in [KKK19]. Instead of taking too much time trying to find these errors, we preferred to try something new.

5.4.2 Improvement

The idea was to use [KKK19] work as an inspiration to get a better result. The paper implementation was not convincing most likely due to some errors in the code or some bad choices of parameters. However the idea of refining the prediction was a good direction. The work of [ZDA19] also pushes towards this same conclusion. They use a 4 stages classification always using binary classification. After the first 3 stages, all patients are classified except MINF and DCM (that is, classes 1 and 2). Therefore, it shows that the difficult work relies on correctly classifying classes 1 and 2.

Since our first classification using a simple Random Forest (n=100 and 14 features) seemed to give better results than the Voting Classifier we chose to use this classifier for the first stage. Once we got this first stage prediction, we wanted to refine it using a binary classifier. There are now two remaining questions, which classifier and what features to choose ?

We first tried some random features without trying any intelligent selection. The results were not convincing. Then we tried to choose the features that could separate the most between both classes. For that, we used 15 in order to find the best features. As we can see, there are a lots of features that have distinct distributions. We chose to only use 4 of them in order to separate clearly both classes: **ESV LV, EDV Myo, ESV Myo, ESV RV**. This is how we motivated that choice: we aim to classify between Myocardial infarction (MINF) and Dilated cardiomyopathy (DCM), and MINF is related to Myocardium whereas DCM is related to volumes (it is a disease that makes ventricles grow larger). We therefore wanted two features that could describe Myocardium, i.e. EDV Myo and ESV Myo. Then we wanted to describe left and right volumes. In order to make sense we thougt that we should describe them at the same instant. Since the plot of ESV LV shows two distinct distributions that are more differentiable than EDV LV (not the same height, mean, and standard deviation), we chose ESV LV to describe LV volumes. The last choice was restrained to ESV RV.



((a)) ESV LV ((b)) EDV RV ((c)) ESV RV

((d)) EDV Myo ((e)) ESV Myo ((f)) EF LV

((g)) EF RV ((h)) RV/LV ED ((i)) RV/LV ES

((j)) Myo/LV ED ((k)) Myo/LV ES ((l)) EDV LV

Figure 15: Distributions of some of the clinical features (class 1 in blue vs class 2 in red)

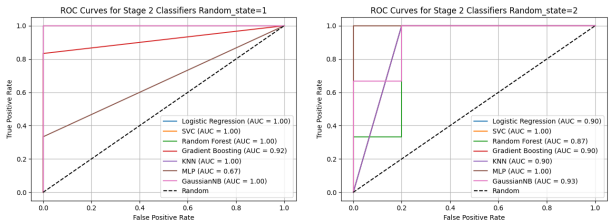
Based on the selected features for the second stage, we trained several binary classifiers to distinguish between MINF and DCM. Models that initially achieved a mean cross-validation accuracy below 85% were further optimized using Bayesian hyperparameter search. All classifiers ultimately achieved relatively high accuracy scores after tuning, except for the MLP, which improved from 72% to only 82%. Given this underperformance, we expected the MLP to yield the weakest results in the full pipeline. Surprisingly, it produced the highest private test accuracy, as shown in Table 3.

This unexpected outcome may be attributed to un-

intentional regularization. The MLP configuration was initially copied from another project, and included restrictive parameters such as early stopping, strong regularization penalties, and a limited validation fraction. These settings likely reduced overfitting during training, unlike the other classifiers that may have overfit due to more permissive configurations.

Furthermore, we observed that classifier performance varied significantly depending on the data split. To assess robustness, we generated ROC curves under different random seeds for stratified sampling. As shown in Figures 16(a) and 16(b), AUC scores fluctuated considerably: in the first split, most classifiers achieved perfect separation (AUC = 1.00), while in the second split, scores were more moderate, ranging from 0.87 to 1.00.

This variability highlights the sensitivity of our models to the limited dataset size and the specific train-test partitioning. It also suggests that some models are more sensitive to sampling variance. These findings show that in small datasets, perfect scores should be interpreted with caution, as they may reflect overfitting or random chance rather than true predictive ability.



((a)) ROC curves (Split 1) ((b)) ROC curves (Split 2)

Figure 16: ROC curves for different classifiers across two random train/test splits

6 Conclusion

In this report, we addressed the ACDC cardiac pathology classification challenge through a progressive implementation of increasingly complex models, inspired by the state of the art. Starting from basic feature-based models and culminating in a two-stage architecture, we evaluated performance both quantitatively and qualitatively.

Model	**Accuracy (%)**
MLP	94
Logistic Regression	88
K-Nearest Neighbors	82
XGBoost	88
Gaussian Naive Bayes	82
Support Vector Classifier (SVC)	82
Random Forest	82

Table 3: Classification accuracy of different models on the test set (private score).

Our experiments highlighted several key points:

- Simple models like Random Forests can achieve high accuracy (up to 88%) with well-engineered features.
- Multi-stage and ensemble approaches require careful feature selection and tuning to avoid overfitting, especially on small datasets.
- Differentiating between class 1 (MINF) and class 2 (DCM) remains the most challenging aspect, as noted in prior work [KKK19; ZDA19].

To summarize the performance of the key approaches tested, we provide the following overview of private test scores:

Method	**Private Accuracy (%)**
XGBoost (12 features)	34
RF (12 features)	71
RF (14 features)	88
Two-stage voting ensemble (20 features)	77
RF + MLP refinement (4 clinical features)	94

Table 4: Summary of private test performance of all main approaches.

While our final accuracy of 94% does not surpass the claimed perfect score in [KKK19], it reflects a

carefully validated pipeline and a well-justified use of features. Future work may focus on improving class 1 vs 2 separability, applying interpretable ML techniques, and incorporating temporal or spatial features from the cine MRI directly. In particular, [ZDA19] proposes a method to keep interpretability of this classification. But this method requires access to full MRIs and not only to ED et ES instant.

Evaluation Metric

Throughout this challenge, we primarily evaluated our models using accuracy, as it is the official competition metric used for ranking submissions. Accuracy is straightforward and intuitive, especially given that the dataset is balanced across the five pathology classes. However, in reality, all classes are not represented equally. People without any signs of cardiac pathology are more often met. Therefore it is clear that our model is skewed due to class balance but it is comprehensible since we work on disease detection. A false positive case is better than a false negative.

Limitations

While our final results are encouraging, several limitations must be acknowledged. The dataset is relatively small, which introduces variability in model performance depending on how data is split during validation. This is particularly evident in the second-stage binary classifiers, where AUC scores varied significantly across random splits. Moreover, segmentation masks for the test set were inferred using heuristic methods, which may introduce noise in feature extraction. Finally, although we focused on well-established clinical features, the exclusion of temporal or spatial dynamics from the cine MRI may limit the model's capacity to capture more complex pathological signatures.

Additional Datasets for Future Work

To improve generalization and reduce overfitting, future work could use additional public cardiac MRI datasets. Some of the mentioned papers used other data sets as MICCAI. Working with them could help improving consistency of our models.

References

[Ber+18] Olivier Bernard et al. "Deep Learning Techniques for Automatic MRI Cardiac Multi-Structures Segmentation and Diagnosis: Is the Problem Solved?" In: *IEEE Transactions on Medical Imaging* 37.11 (2018), pp. 2514–2525. DOI: 10.1109/TMI.2018.2837502.

[Ise+18] Fabian Isensee et al. "Automatic Cardiac Disease Assessment on cine-MRI via Time-Series Segmentation and Domain Specific Features". In: *Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS Challenges*. Ed. by Mihaela Pop et al. Cham: Springer International Publishing, 2018, pp. 120–129. ISBN: 978-3-319-75541-0.

[KAK18] Mahendra Khened, Varghese Alex, and Ganapathy Krishnamurthi. "Densely Connected Fully Convolutional Network for Short-Axis Cardiac Cine MR Image Segmentation and Heart Diagnosis Using Random Forest". In: *Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS Challenges*. Ed. by Mihaela Pop et al. Cham: Springer International Publishing, 2018, pp. 140–151. ISBN: 978-3-319-75541-0.

[Wol+18] Jelmer M. Wolterink et al. "Automatic Segmentation and Disease Classification Using Cardiac Cine MR Images". In: *Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS Challenges*. Ed. by Mihaela Pop et al. Cham: Springer International Publishing, 2018, pp. 101–110. ISBN: 978-3-319-75541-0.

[KKK19] Mahendra Khened, Varghese Alex Kollerathu, and Ganapathy Krishnamurthi. "Fully convolutional multi-scale residual DenseNets for cardiac segmentation and automated cardiac diagnosis using ensemble of classifiers". In: *Medical Image Analysis* 51 (2019), pp. 21–45.

[ZDA19] Qiao Zheng, Hervé Delingette, and Nicholas Ayache. "Explainable cardiac pathology classification on cine MRI with motion characterization by semi-supervised learning of apparent flow". In: *Medical Image Analysis* 56 (2019), pp. 80–95.