Final Report: IMA205 Kaggle Challenge 2023 on Cardiac Pathology Prediction

Presented by

Nicolas BOISSEAU

Student at Institut Polytechnique de Paris in M2 Biomedical Engineering

IMA205: Machine Learning for image and object recognition

Due for the 07 May 2023

INSTITUT POLYTECHNIQUE DE PARIS

Table of content

Table of illustrations

1. Challenge introduction, rules and constraints:

Regarding the challenge information, the guidance and constraints have been thoroughly described in the Kaggle challenge webpage. Here I'm only mentioning the description from: https://www.kaggle.com/competitions/ima205-challenge-2023/overview

1.1. Introduction

Analysis of cardiac function is essential in clinical cardiology for disease diagnosis, patient management and therapy decision. The challenge focuses on four cardiac pathologies that might go unnoticed at first, but can ultimately become life-threatening. Complications include heart failure and sudden cardiac arrest.

It is important to identify these conditions as early as possible to guide treatment and prevent complications. This is why several approaches for automatic diagnosis from cardiac magnetic resonance imaging (CMRI) have been proposed in the last years (non-invasive computer-aided diagnosis (CAD)).

1.2. Goal of the challenge

The goal of this challenge is to classify MRI images of the heart among five different diagnostic classes:

1. Healthy controls
2. Myocardial infarction
3. Dilated cardiomyopathy
4. Hypertrophic cardiomyopathy
5. Abnormal right ventricle

In order to do so, you will extract features such as the volume of the anatomical structures at two different time points in the cardiac cycle (end of the cardiac contraction and end of the dilation), the thickness of the cardiac muscle, and the ejection fractions. After that, you will use machine learning algorithms to classify the subjects.

1.3. Data

You will use a dataset of 150 subjects with their MRI images and, when available, their corresponding segmentations and metadata (subject height and weight). Data has already been randomly split into a training-validation set (two thirds = 100 subjects) and a test set (one third = 50 subjects). You only have the classification (made by clinicians) of the training-validation set. The goal of the challenge is to estimate the correct class of each subject in the test set. **You can only use the data provided in this challenge**

The class labels ("Category" field in the metadata) are mapped thusly to diagnostic classes:

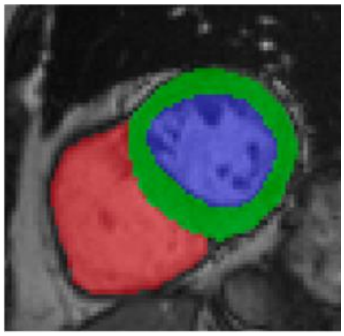
'0' - Healthy control
'1' - Myocardial infarction
'2' - Dilated cardiomyopathy
'3' - Hypertrophic cardiomyopathy
'4' - Abnormal right ventricle

For each subject, two MRI images are provided : one image at end diastole (end of dilation in the cardiac cycle) and one image at end systole (end of contraction). Each MRI image is a 3D volume containing the heart and adjacent structures. Comparing the cardiac anatomy between these two time points should help you classify subjects.

To help you achieve your goal, you are provided for each image in the training-validation set with a corresponding 3D segmentation of the cardiac anatomy in three substructures: left ventricle cavity, right ventricle cavity, and myocardium (cardiac muscle). Each segmentation map consists in a 3D multi-label mask with the corresponding labels:

0 - Background
1 - Right ventricle cavity
2 - Myocardium
3 - Left ventricle cavity

2D slice of CMR image



Left ventricle cavity, myocardium, right ventricle cavity & background

You can use these segmentations to extract relevant features about the subject's cardiac anatomy. **In the test set, you are only given partial segmentations.** Specifically, the left ventricle cavity label (3) is missing (replaced by background). Therefore you may find useful as an intermediate step to perform left ventricle segmentation, using any suitable computer vision or machine learning technique.

1.4. Feature extraction

You can use all features you would like. A list of references describing very well-known features can be found at the end of this page but you can find many more articles in the literature. You can use Pubmed, Google Scholar or simply Google to look for them. By using the network of Télécom Paris or its VPN, you will have automatically access to most of the scientific journals.

1.5. Evaluation

The evaluation of the challenge will be based on 1) the ranking in the leaderboard, 2) a report and 3) the quality of your code.

The students who will have the best rankings in the (private) leaderboard at the end of the challenge will have between 1 and 4 points more in the final grade (depending on the result).

You will have to write a report where you will thoroughly explain the extraction of the features, the classification algorithms you used, why you chose them and the potential pre- and post-processing (such as the LV cavity segmentation). You will have to explain the results saying if and why you expected those results. **Be careful ! You will be penalised if you simply do a list of results like "I tried this algorithm but it did not work so I tried another one and so forth ..." !!**

Write a proper, commented and clean code in Python 3. We will test it and if it does not work you will have a penalty on the grade. Please write at the beginning of your code the version of the libraries you used. You can write a set of functions with a main one - we will only run the main function - or a jupyter-notebook.

Everything, report and code, must be uploaded to E-campus in the section Challenge before the end of the challenge

1.6. Metrics

As ranking metric, we will use the Categorization Accuracy, which is defined as:

$$Acc = \frac{1}{N}\sum_{i=1}^{n} I(y_i = f_i)$$

where N is the number of test subjects, y_i the ground truth label for subject i, and f_i the predicted label.

1.7. Methods and software

You must use Python 3 as programming language. We strongly suggest that you use numpy and scikit-learn.

You can use any pre-processing and post-processing - coded by you or correctly referenced in the report.

All CMR images and segmentation maps in the dataset are saved as NIfTI files (.nii extensions). You can read/write such images using one of several python libraries, such as nibabel or TorchIO.

To visualize MRI images and segmentations, you may directly display 2D slices in a jupyter notebook using matplotlib after loading the image, or you may use dedicated visualization software such as ParaView (after loading the AnalyzeNIfTIReaderWriter plugin), 3D Slicer and others.

1.8. References

For the (optional) left ventricle segmentation:

- https://ieeexplore.ieee.org/document/4776520
- https://www.sciencedirect.com/science/article/pii/S136184150400026X
- https://www.sciencedirect.com/science/article/abs/pii/S0010482505000430
- https://link.springer.com/chapter/10.1007/978-3-642-31196-3_3#chapter-info
- https://link.springer.com/chapter/10.1007/978-3-642-28326-0_11#chapter-info
- https://link.springer.com/chapter/10.1007/978-3-319-10470-6_73#chapter-info

For the classification:

- https://link.springer.com/chapter/10.1007/978-3-319-75541-0_15
- https://link.springer.com/chapter/10.1007/978-3-319-75541-0_13
- https://link.springer.com/chapter/10.1007/978-3-319-75541-0_11

2. Cardiac pathology knowledge

2.1. Myocardial infarction

I read the disease information on https://www.ncbi.nlm.nih.gov/books/NBK537076/

Myocardial infarction is a common and serious cardiac pathology that results from the obstruction of one or more coronary arteries, leading to a lack of oxygen and nutrient supply to the heart muscle. Myocardial infarction can cause permanent damage to the heart muscle and can lead to heart failure, cardiac arrest, and even death.

To diagnose Myocardial infarction, physicians typically use a combination of symptoms, electrocardiography (ECG) results, and blood tests for cardiac biomarkers such as troponin.

For classification purposes, the features of interest are:

- abnormal myocardial wall thickness (myocardium cavity)
- LV dysfunction (LV cavity)
- LV dilation. (LV cavity)

2.2. Dilated cardiomyopathy

https://www.ncbi.nlm.nih.gov/books/NBK441911/

Dilated cardiomyopathy is a condition in which the heart becomes enlarged and weakened, leading to reduced cardiac output and heart failure. Dilated cardiomyopathy can be caused by genetic mutations, viral infections, or exposure to certain drugs or toxins. Symptoms may include shortness of breath, fatigue, and swelling in the legs and feet. For classification purposes, features of interest include:

- LV dilation (LV cavity)
- LV dysfunction (LV cavity)
- RV dilation. (RV cavity)

2.3. Hypertrophic cardiomyopathy

https://www.ncbi.nlm.nih.gov/books/NBK430820/

Hypertrophic cardiomyopathy is a genetic disorder that affects the heart muscle and causes it to thicken, making it harder for the heart to pump blood effectively. Hypertrophic cardiomyopathy can cause symptoms such as shortness of breath, chest pain, and fainting. It can also lead to sudden cardiac arrest, especially in young athletes.

For classification purposes, the features of interest for Hypertrophic cardiomyopathy include:

- abnormal myocardial wall thickness (myocardium cavity)
- LV dysfunction (LV cavity)
- LV outflow tract obstruction. (LV ejection rate)

2.4. Abnormal right ventricle

One abnormal right ventricle example is the right ventricle hypertrophy, https://www.ncbi.nlm.nih.gov/books/NBK499876/

Abnormal right ventricle (ARV) can result from a variety of cardiac pathologies, including pulmonary hypertension, RV hypertrophy, and RV dysfunction. ARV can lead to symptoms such as shortness of breath, chest pain, and fatigue.

For classification purposes, features of interest for ARV include:

- RV dilation (RV cavity)
- RV dysfunction (RV cavity)

3. Data exploration and visualization

As a novice in Machine Learning, I wanted to first observe the data and the files available for the cardiac pathology prediction. This help me to organize on what should be done.

I started by downloading the data set on my drive and I made a first notebook cell in order to visualize a random patient images from the training set. The first row of images corresponds to the end diastole volume, the second the end diastole segmented, the third the end systole volume and the fourth the end systole segmented. There are several slices of images corresponding to the 3D representation of the heart MRI.

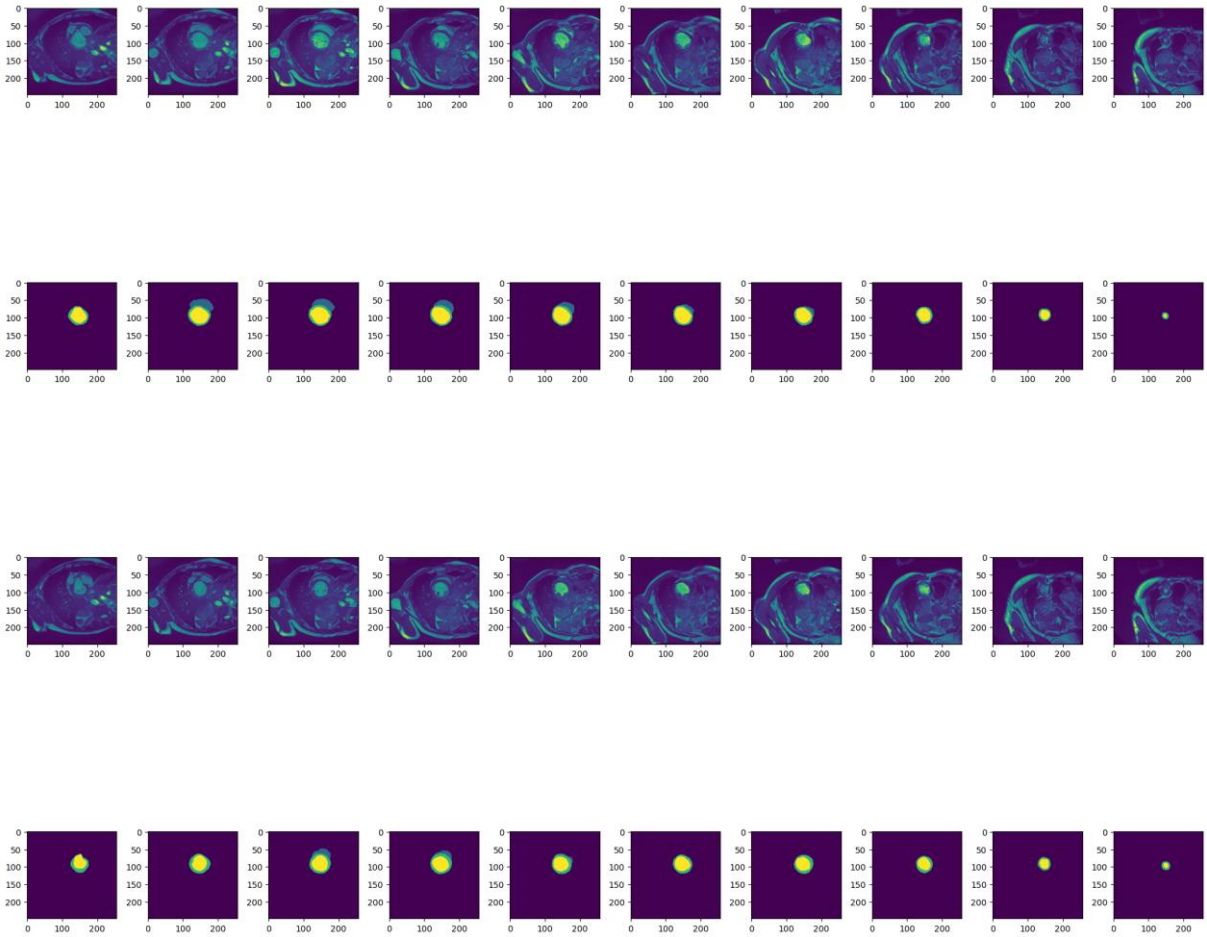


Figure 1: Training set exploration of a random subject

I plotted an histogram to display the variability of the images sizes in the training set. And as expected the images sizes were different not only on the slices. So this information was useful to gather and needed to take into account for the future feature extraction part.

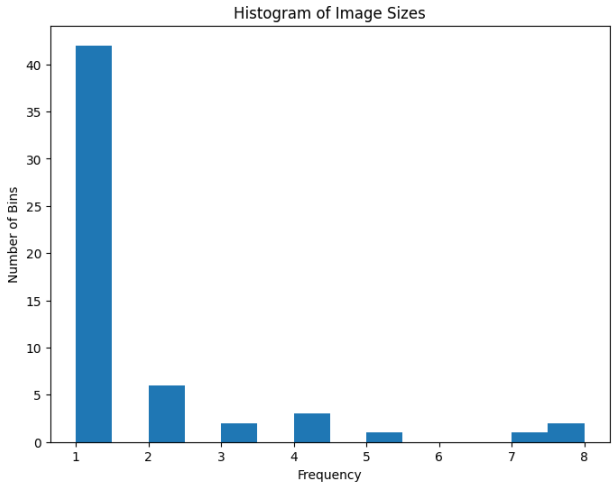


Figure 2: Image sizes variability

I also open the metadata files on load them as dataframe to observe their shape and content. Then I plot the diseases categories of the training set because I wanted to make sure I don't have any unbalance dataset problem that would lead to bias in the classification. However, the subjects weren't shuffle.

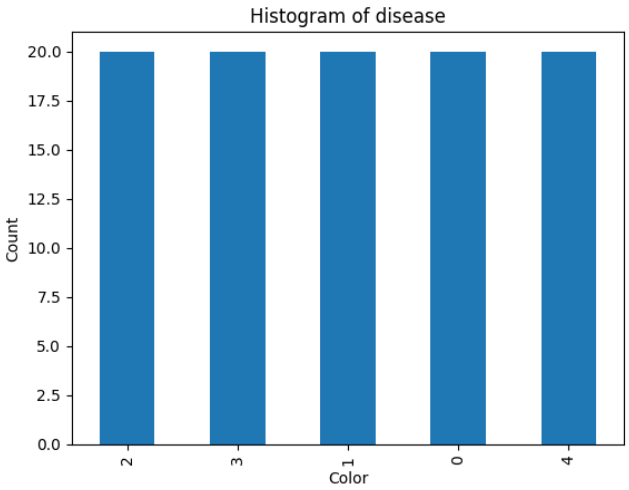


Figure 3: Diseases distribution

4. Left ventricle cavity segmentation

4.1. Prior test set exploration

Prior feature extraction, a necessary part was to segment the left ventricle cavity missing in the test set. It was necessary because as seen in the diseases descriptions, the left ventricle structure account for a significant amount of times as a region of interest where features should be extracted.

While exploring the test set and comparing with the training set, I could quickly observe and realize that the missing LV cavity that should be segmented is always inside the myocardium muscle.

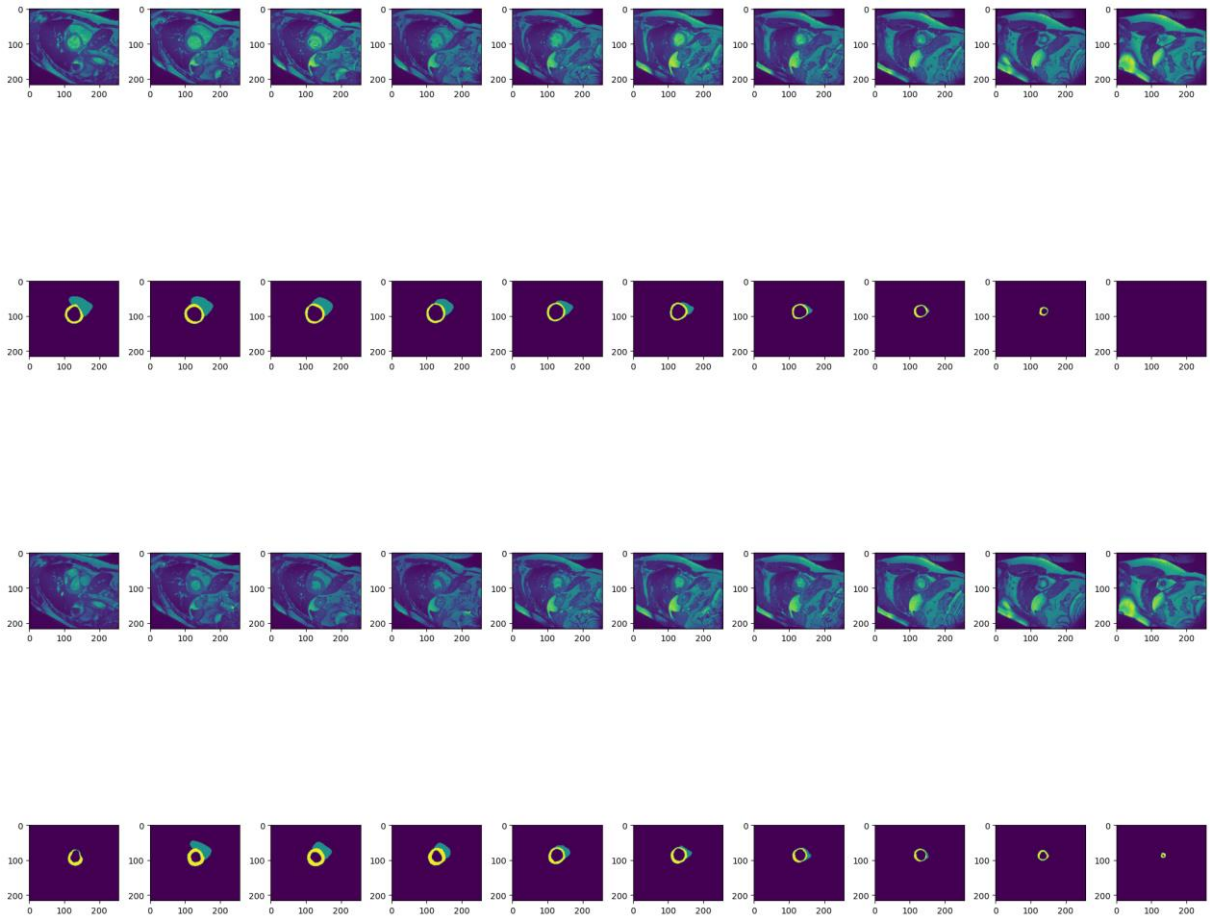


Figure 4: Test set exploration with a random subject

4.2. Snake expansion segmentation

With the knowledge I gained, I implemented two different methods to segment the left ventricle (LV) cavity: snake expansion and flood filling.

First, I developed a function to identify the centers of the myocardium on each image slices. For the snake expansion method, I utilized the active contour technique that I learned in the IMA204 class. However, achieving accurate results with this method proved to be time-consuming as I needed to carefully adjust the initial parameters and the expansion process.

Snake expansion segmentation relies on the idea of using active contours to iteratively deform an initial contour to fit the boundaries of the LV cavity. By optimizing energy functions that consider internal forces (e.g., smoothness, elasticity) and external forces (e.g., image gradients, intensities), the contour gradually expands to accurately outline the LV cavity boundaries.

Although the snake expansion method offers advantages in capturing complex shapes and adapting to different image characteristics, finding the right balance between forces and initial contour placement was challenging. It required meticulous parameter tuning.

While initially encountering difficulties in achieving accurate segmentation and fine-tuning the parameters, I eventually obtained promising results by using the flood filling method.

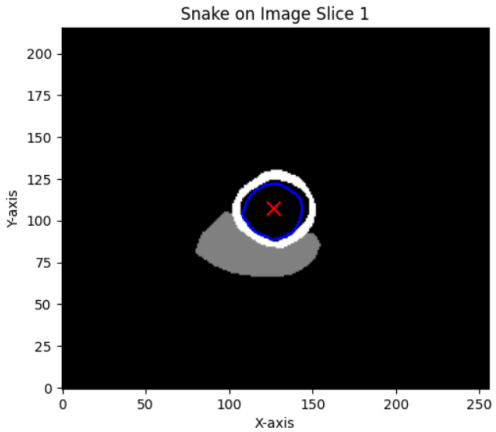


Figure 5: Snake expansion for LV cavity segmentation

4.3. Flood filling segmentation

I explored alternative methods to segment the left ventricle (LV) cavity, specifically focusing on filling the connected pixels within the myocardium centers. During my research, I came across an example of a flood fill function in the scikit-image library https://scikitimage.org/docs/stable/auto_examples/segmentation/plot_floodfill.html. Intrigued by its potential, I decided to apply this technique and observe its impact on the segmentation results.

The flood fill method operates by starting from a seed point within the myocardium and progressively expanding to neighboring pixels with similar intensities until a stopping condition is met. This iterative process effectively fills the connected region of interest, generating a complete segmentation of the LV cavity.

By implementing the flood fill segmentation approach, I noticed significant improvements in the accuracy of the segmentations. This technique proved to be particularly advantageous in scenarios where the myocardium exhibited irregular shapes.

It is worth mentioning that the flood fill method also introduced computational efficiency compared to the snake expansion technique. The simplicity and effectiveness of this approach made it a viable alternative for LV cavity segmentation, facilitating the subsequent feature extraction process.

Overall, the implementation of the flood filling segmentation technique provided valuable insights into alternative methods for LV cavity segmentation. Its ability to fill connected pixels within the myocardium resulted in improved segmentation accuracy, contributing to more reliable feature extraction and subsequent cardiac pathology classification.

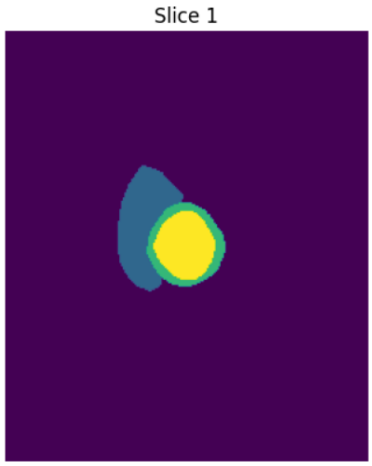


Figure 6: Flood fill method for LV cavity segmentation

4.4. Post segmentation, test set exploration

Then I loop through all the image of the test set in order to generate the LV cavity segmentation. I named the generated images with the following names:

- XXX_LV_ED_seg.nii
- XXX_LV_ES_seg.nii

To check visually my result, I displayed several times a random subject segmentation pictures and compare them to the one from the training set.

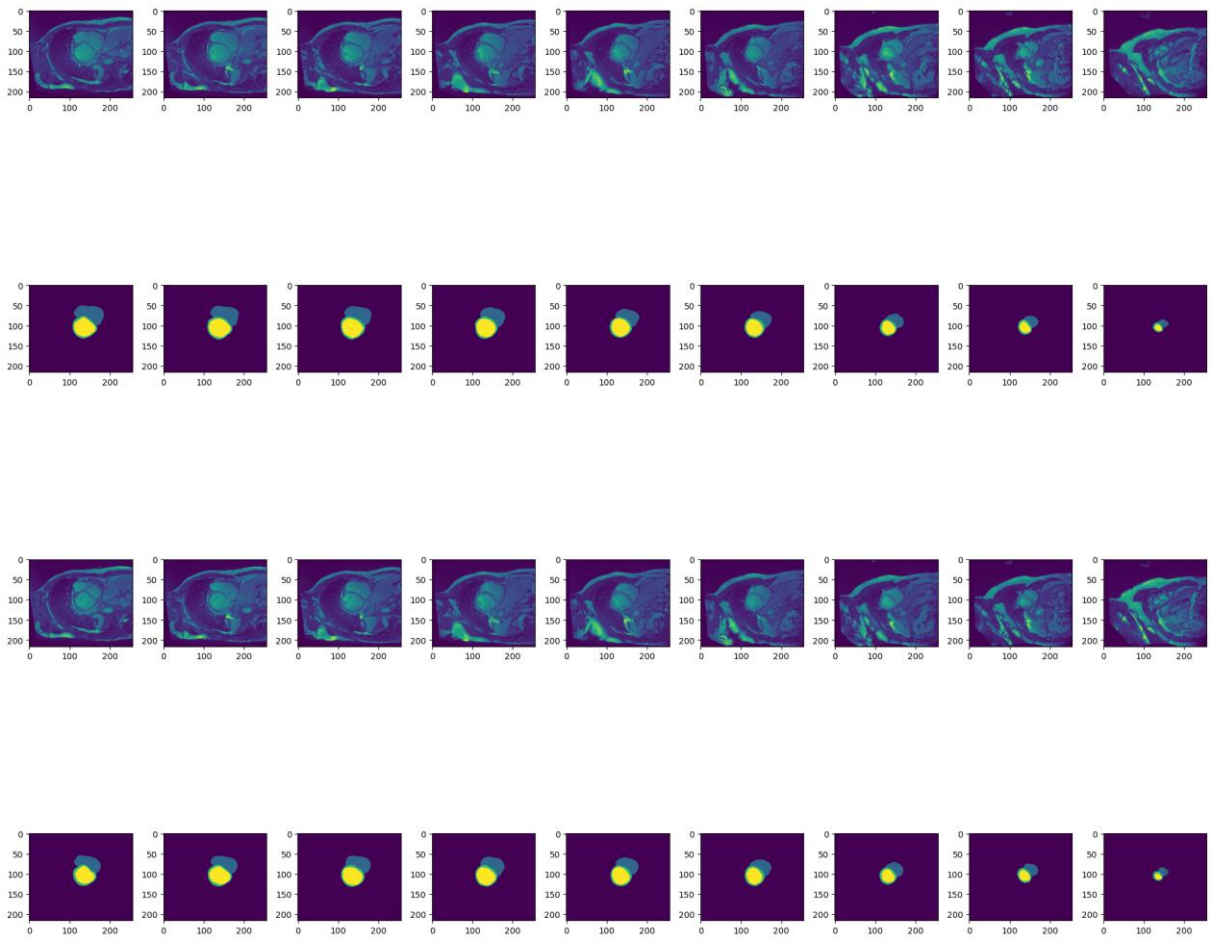


Figure 7 : Test set heart MRI images after LV cavity segmentation

5. Heart MRI segmented image features extraction

5.1. Extracted features choice.

Once I have access to the LV cavity segmentation on the test set. I can now perform extraction features on all the data sets. Reading the articles at our disposal and looking for more information online, I decided to extract the following features based on this article https://link.springer.com/chapter/10.1007/978-3-319-75541-0_11 :

- 1: volume of the left ventricle at the end of diastole (lv_vol_ed)
- 2: volume of the left ventricle at the end of systole (lv_vol_es)
- 3: volume of the right ventricle at the end of diastole (rv_vol_ed)
- 4: volume of the right ventricle at the end of systole (rv_vol_es)
- 5: end of diastole myocardium volume (myo_vol_ed)
- 6: end of systole myocardium volume (myo_vol_es)
- 7: ejection fractions of left ventricle cavity (lv_ef)
- 8: ejection fractions of right ventricle cavity (rv_ef)
- 9: ratio between RV and LV volume at end diastole (r_rv_lv_ed)
- 10: ratio between RV and LV volume at end systole (r_rv_lv_es)
- 11: ratio between Myocardium and LV volume at end diastole (r_myo_lv_ed)
- 12: ratio between Myocardium and LV volume at end systole (r_myo_lv_es)
- 13: body mass index (bmi)

5.2. Features details

The first four features (1,2,3,4) represent the volume of the left and right ventricles at the end of diastole and systole, respectively. These volumes are important indicators of heart function and can be used to assess the size of the ventricles.

The features (5,6) are the end of diastole and systole myocardial volumes. These volumes represent the myocardium thickness present at the end of each cardiac cycle.

The ejection fractions of the left and right ventricles, respectively (7,8) represents the percentage of blood pumped out of the ventricles during each heartbeat and is an important measure of heart function as well.

The ratio between the RV and LV volume at end diastole and end systole provides information on the relative size of the two ventricles (9,10). The ratio between the myocardium and LV volume at end diastole and end systole (11,12) represents the proportion of myocardium relative to the LV volume.

Finally, the last feature is the body mass index (13), which is a measure of body fat based on height and weight. It is not included in the feature of the study. However I could clearly see that the height and weight were not accounting significantly in the feature importance so I decided to use the bmi as it has been shown to be a risk factor for heart disease. Indeed my end results classification was improve by using bmi instead of height and weight.

5.3. No dimensionality reduction

I decided not to implement dimensionality reduction methods such as KPCA, ICA, and PCA. Since I had a relatively small number of features and subjects, which made it unnecessary to apply these methods. These methods are generally useful when dealing with large datasets with many features, which can result in increased computational complexity. In Additionally, I wanted to keep the interpretability of my model.

5.4. Features importances

While doing my cross validation, I'm able to see the ranking of the importance of the extracted features:

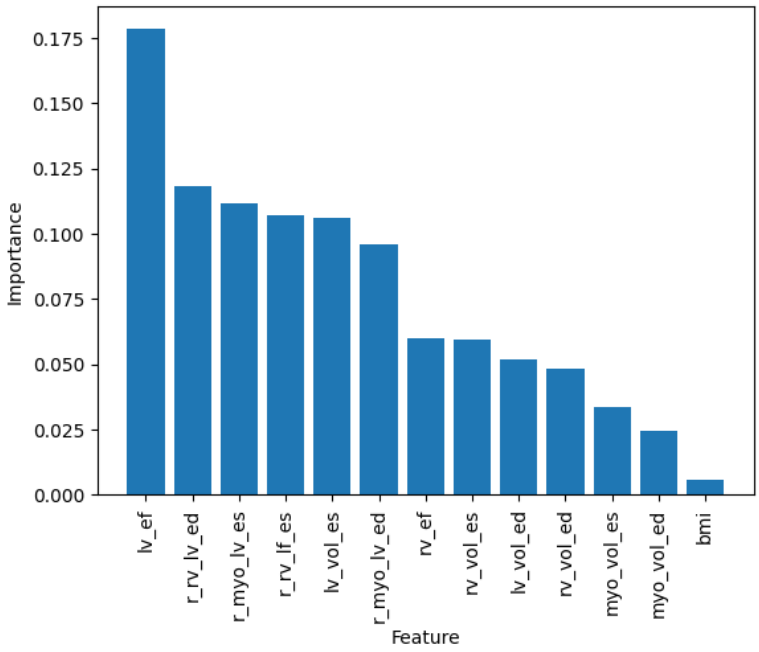


Figure 8: Final extracted features importances with bmi

I actually use this chart to improve my extraction features through the project, since at first, I missed some interesting features. At the end I also realize that the weight and height are really poor, and I decide to try to use body mass index and it improved slightly the classification. Those charts proved the significant impact that features extraction have on the classification results. Once again, information is key.

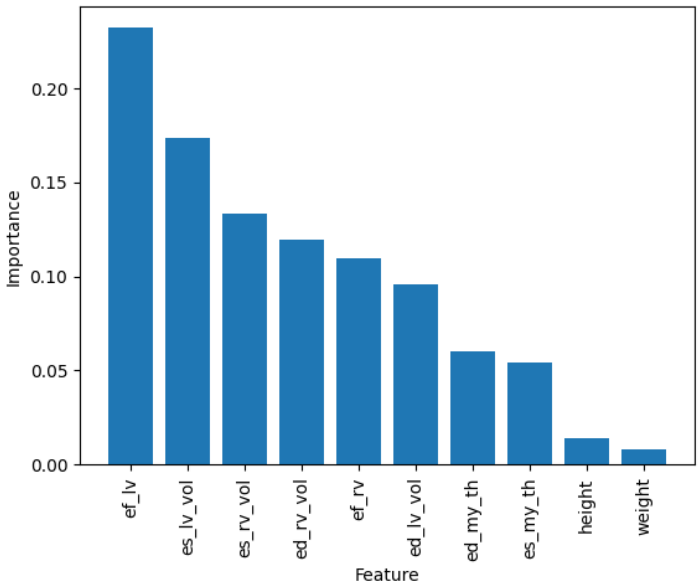


Figure 9: Initial extracted features importances

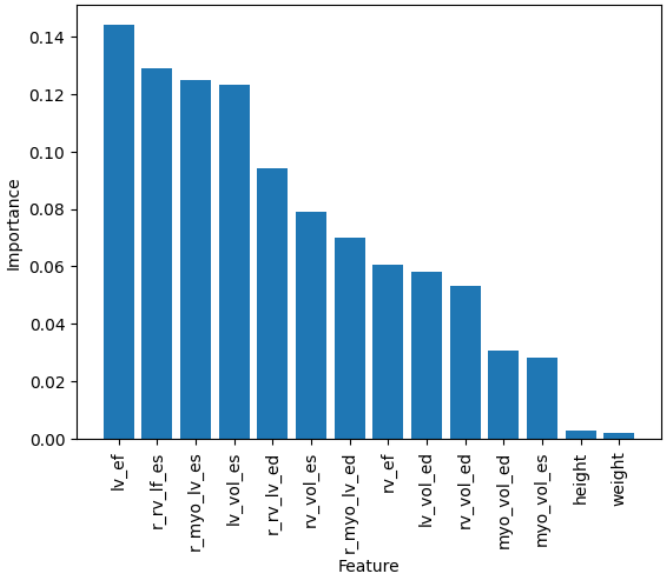


Figure 10: Extracted features importances with weight and height

6. Classification approach for MRI cardiac diseases prediction

6.1. Data organization

Since I saved my features for the training and test set in csv files I have to reload them remove some columns such as the ID or the Category in order to organize my data in X_train, y_train, and X_test.

6.2. Training set shuffle

After organizing I had to shuffle my training set since when I observed the metadataTrain csv file, the subject was listed by category. In the purpose of mixing the category and avoiding bias in the model training I shuffled the training set and training label accordingly.

6.3. Features normalization

One important step is to rescale or normalize the data. So, It is done to ensure each feature will have the same influence as another one and bring them to a same level of importance before the model training.

6.4. Random Forest Classification

For the cardiac pathology classification, I choose to implement Random Forest Classifier for several reasons listed here after:

- RFC can capture non-linear relationships between features. Complex interactions between features such as left ventricular volume, right ventricular volume, and ejection fraction may not be easily captured by simpler models. RFC in other hand is able to capture these complex relationships by creating decision trees that partition the feature space in a non-linear way which I'm definitely interested in this classification task.

- RFC is also robust to noisy data and missing values, which was happening at the beginning of my LV cavity segmentation with active contour and without it. RFC is able to handle these issues by training on multiple decision trees and aggregating the results. This leads to a more robust classification.

- Interpretability: RFC produces a measure of feature importance, which allows me to interpret which features are most important for the classification task. This is useful for me in order to improve the features I selected for my classification and to see the poorly relevant one. In fact, the end accuracy results directly improved by my extraction features choice.

- Ensemble learning: RFC is an ensemble learning method, which means that it combines the predictions of multiple decision trees to produce a final prediction. This can increase the accuracy and robustness of the model by reducing the impact of individual decision trees that may be biased or overfit to the data.

For all those reasons, non-linearity capture, noise robustness, interpretability and ensemble learning, I decided to use a RFC model to do the classification.

6.5. Best parameters selection with cross-validation

In order to optimize the RFC model parameters I split my training set in training set 2 and validation set with a ratio 80%, 20%. Then I used cross validation on those data to retrieve the best possible parameters for the best accuracy. The parameters are the *n_estimators*, the *min_sample_leaf* and the *max_features*.. Then I fit my whole RFC model with the optimize parameters to the entire training set.

6.6. Classification prediction

Once the RFC model is fitted with the training set, I predict the y_test label by passing the test set in argument. I save the predicted label such as the submission file and submit it to the Kaggle challenge form manually.

7. Submission results

During this competition, I submitted mainly three different versions of my code, each resulting in three varying performances. The differences among these versions were the features I selected for the classification task.

My first implementation, I used only 10 basic features related to the sum of the heart structures to calculate volumes. This initial approach achieved a test score of 0.666, indicating moderate performance.

To improve my results, I conducted further research on cardiomyopathy article available in the challenge description and I identified more efficient features. Including these new features into my model led to a significant improvement in performance, with a test score of 0.800.

Additionally, I analyzed the importance of features and identified that the height and weight variables had limited impact on the classification. Considering this, I decided to leverage the body mass index (BMI) as a superior indicator, which not only simplified the model but also enhanced its predictive capabilities. With this modification, I achieved a higher test score of 0.866.

These results demonstrate the importance of feature selection. By incorporating more relevant features, I was able to achieve better classification performance.

8. Additional improvement : Data augmentation

During the competition, one area that I would have liked to explore further to improve the performance of my model is data generation techniques. Given the limited size of the training dataset, data generation methods could have been beneficial in enhancing the robustness and generalization capabilities of my model.

I mostly wanted to implement data augmentation, which involves applying various transformations to the existing data images to create bigger samples. Techniques such as rotation, translation, scaling, and flipping would have introduce diversity to the dataset, allowing the model to learn more representative and invariant features. Data augmentation could have help overcome overfitting and improve the model.

By incorporating data generation techniques, I could have potentially increased the size and diversity of the training dataset, leading to better model performance and improved classification accuracy. Unfortunately, I was unable to explore data augmentation techniques within the given time constraints, I recognize their potential in enhancing the performance of the model and would consider them as a valuable for future improvement in similar projects.

9. Conclusion

In this project, as a biomedical student, I developed an automated cardiac diagnosis system using a dataset of cardiac MRI images. The project involved several key steps, including LV segmentation, feature extraction, data preprocessing, classification, and analysis. By processing techniques, I successfully segmented the left ventricle (LV) from cardiac images, allowing for accurate measurement of LV-related features such as ejection fraction and ventricle volumes. Through data analysis and visualization, I gained valuable insights into the variations of these features across different cardiac conditions. While time constraints limited the implementation of certain enhancements, such as data augmentation, the project has provided me with a solid foundation in applying feature extraction, data analysis, visualization, and machine learning techniques in the field of cardiac healthcare. This project not only demonstrates the potential of automated cardiac diagnosis but also highlights the practical relevance and interest for me as a biomedical student. It will definitely serve me as a steppingstone for further research project and biomedical field.