Challenge IMA 205 - Report

BAHOUS Youness

7 mai 2023

Leaderboard username : bhsyns

Introduction

Heart disease is a leading cause of death worldwide,

Early detection and diagnosis can significantly improve patient outcomes. Therefore, accurate classification of heart disease is crucial for effective treatment and management. Techniques of classifiction such as these studied on IMA205 have shown promising results in medical diagnosis and decision-making. In this Kaggle challenge the main task was to successfully classify patients into 5 classes , It offers as well an optional segmentation task that involves identifying the left ventricle of the test subjects , this task was rendered easy by the fact that we already have the segmentations the right ventricle cavity and the myocardium. The objective of this challenge is to provide healthcare professionals with reliable and efficient tools for diagnosing heart disease and improving patient outcomes. In this report, we present the methodology, results, and analysis of our approach to this AI challenge.

In this report , the methods of classification used and tried for this challenge are exposed to obtain 86.6 % score on the leaderboard.

1 Data Preparation

The data for this challenge consists of 150 scans for 150 patients the scans were in 2 NIfTI files one scan on ED phase and the other on ES phase . segmentations were as well provided for 4 regions (counting the background)in the case train subjects and for 3 regions of the test subjects only for two regions. this data was organised in folders

To read these files I used the nibabel package . and loaded the training and testing examples separately so as to be able to access them more effeciently (Code in the notebook)

In order to perform the segmentation task , I needed to extract images both from the train data and test data to work on each slice separately . I have done this for two purpuses :

1. Prepare a training and testing set to use when training a convolutional neural network based segmentation method
2. Use the naïve segmentation method specified below

I also did extract the voxel size from the header since it will be used to compute some of the features taken into consideration.

2 Segmentation

As mentioned on the introduction

Throughout the time of the challenge I have tried multiple ideas for the segmentation of the left ventricle on the test set :

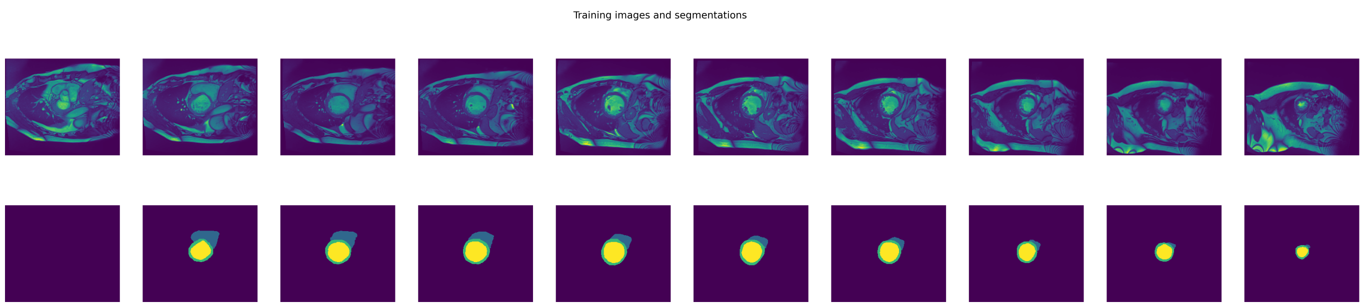


Figure 1 – A plot showing different slices and their corresponding given segmentations for a patient in the training dataset

2.1 Naive approach based on provided myocardium segmentations :

By noticing that the left ventricle in most slices corresponds to the interior of the myocardium . A first Idea that comes to mind is using the the segmentations that we already have and fill the inside of the myocardium as an estimation of the left ventricle.

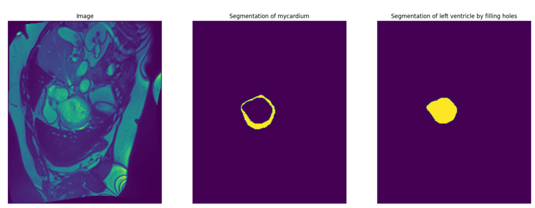


Figure 2 – Architecture of convolutional neural network used for lv segmentation (taken from [1])

There are two major iimmediate problems with using this approach

— The left ventricle doesn't always correspond to the inside of the myocardium , and this can bias the computations of volumes , and consequently the decision of the classification model
— In real world applications we do not have the myocardium segmentations , and if our desire is to create an automated pipline this method is too naïve to be used

Hence I thought about using another approach

2.2 Segmentation using a UNET Convolutional neural network :

To apply some of the skills acquired in this course I implemented a CNN for segmenting the test images . Noticing that the slices of 3d segmented and non segmented volumes from the training data of the patients provide a good database with over 2000 images , I chose to work with a Unit as proposed in [1] , it has the architecture as the one given In the chart below , In the following chart N channels are replaced by only one channel , since I worked on 2D slices , with segmentations of the lv only , since we already have the segmentations for the other regions.

To prepare the training dataset , I extracted the slices and lv segmentations as slices from these training images . I extracted 17% to perform model validation and to . scaled , and augmented with rotations in range 0.2 radians . and forced them to a unified size of 256×256.

for the loss function I used the binary cross entropy loss , and for the optimizer I used the Adam optimizer .

The model was trained on 80% of the data and validated on 20% of the data . The model was trained for 100 epochs with a batch size of 32 . The model was trained on a Colab GPU . To avoid overfitting I used the early stopping callback with a patience of 10 epochs on the loss function .

Here are plots of both the losses and validations during the model training

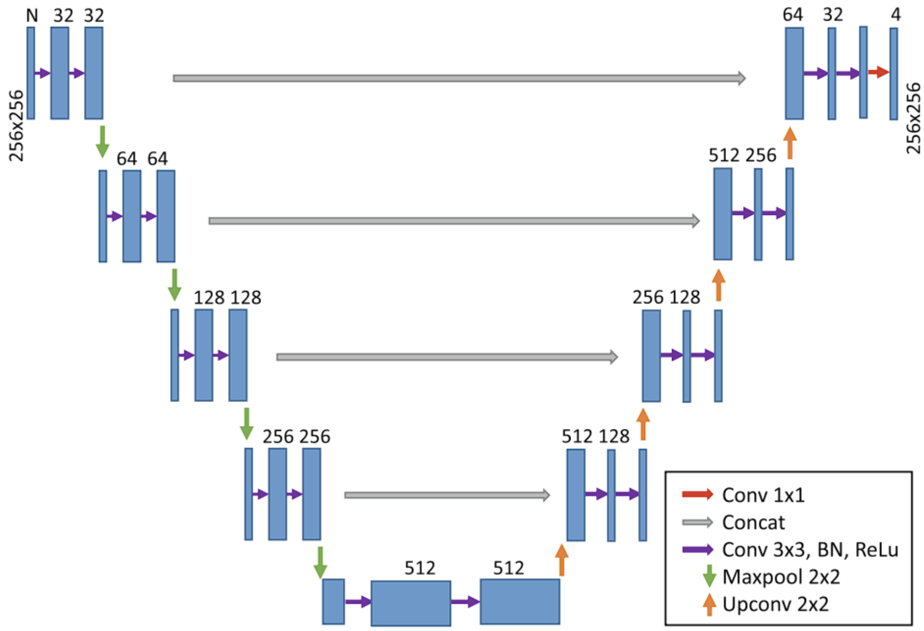
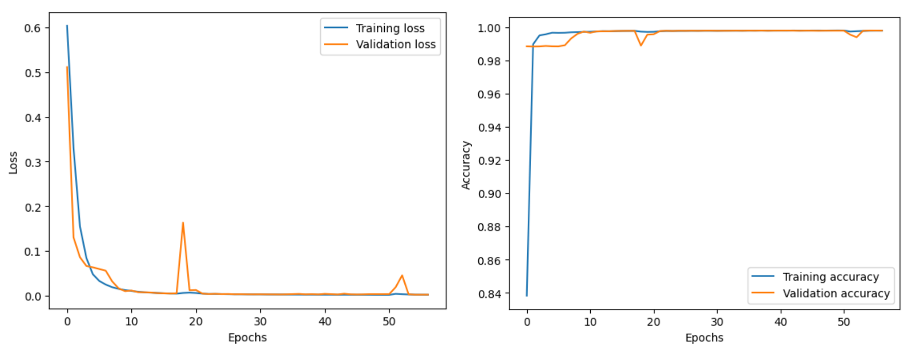


Figure 3 – Architecture of convolutional neural network used for lv segmentation (taken from [1])



(a) Losses (b) Accuracy

Figure 4 – Unet segmenattion Training loss and accuracy evolution

The code for this part is in a Colab notebook , since it was the most conveinient format to perform the training .

Link for the segmentation code : Colab Notebook Link

The model was saved to be used in the classification code on the test set image slices.

3 Classification

The main task of the challenge is the classification of the patients into the following 5 categories , The main challenge with the given dataset is choosing the right features , since the amount data in our posession is small . we only have results for 100 patients

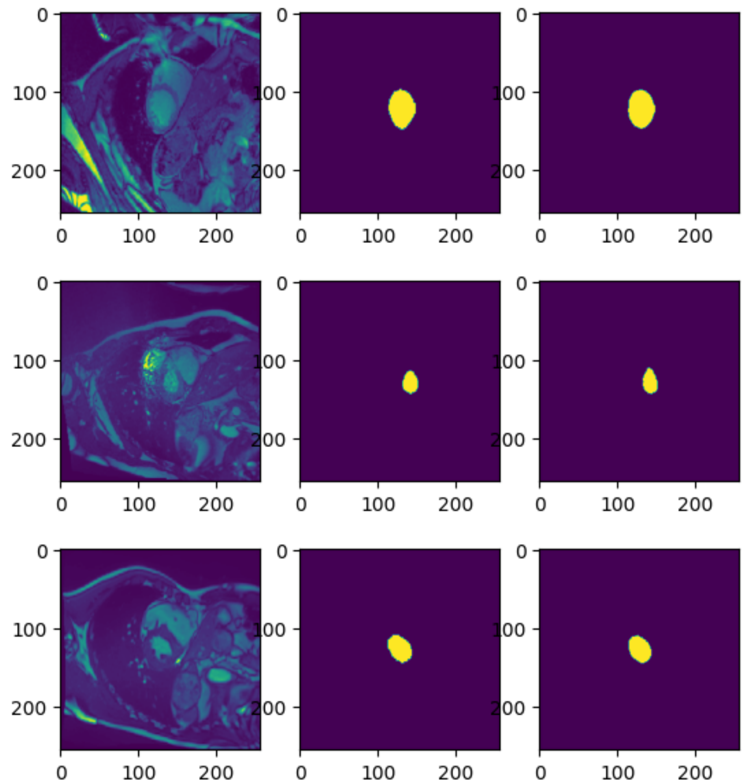


Figure 5 – Some of the segmentations obtained using Unet on unseen validation data. **To the left :** The slice , **In the middle** : The given segmentation , **To the right** : The result using Unet segmentation

3.1 Features

All of the features I used were extracted from the segmentations and the metadata of the patients , these features were inspired by papers on this classification task [2],

Here is a table of the features I used in my classification :

— For each of the model I tried to compute SHAP values and Importance weights to interpret the importance of each of the features in the classification and how it impacts the decision. And I chose to keep the random forrest classifier since it was the one that made more sense in terms of how it used the provided features.

3.2 Classification algorithms :

For the classification I tried many methods , Tree based methods , and Kernel Based ones , and I also used simple MLP architecture for the classification . My final choice for the model took into account

3.2.1 Random Forest classifier

Tree based methods are best suited for classification tasks. The reason being, every tree grown in a random forest algorithm is independent of the other.

That is, each tree is grown by using a different set of observations and different set of features. The final prediction can be obtained by aggregating the predictions of all the individual trees. In this way, the final prediction will be based on all the features in the dataset, and the final model will be more robust. And they are also very easy to use and interpret. They also provide a pretty good indicator of

Feature N°	Feature name
1	Patient weight
2	Patient hight
3	Index of mass
4	Body Surface Area (via mosteller's formula)
5	left ventricle volume at end of the dilation
6	right ventricle volume at end of the dilation
7	myocardium volume at end of the dilation
8	$\frac{\text{myocardium volume}}{\text{right ventricle volume}}$ at end of the dilation
9	$\frac{\text{myocardium volume}}{\text{left ventricle volume}}$ at end of the dilation
10	left ventricle volume at end of the contraction
11	right ventricle volume at end of the contraction
12	myocardium volume at end of the contraction
13	$\frac{\text{myocardium volume}}{\text{right ventricle volume}}$ at end of the contraction
14	$\frac{\text{myocardium volume}}{\text{left ventricle volume}}$ at end of the contraction
15	Ejection fraction of left ventricle
16	Ejection fraction of right ventricle

Figure 6 – Table of the features used in the classification

the importance it assigns to your features. as we can see using shap values. The first model I used is a Random Forest classifier. I used a grid search to find the best hyperparameters for the model.

The hyperparameters I tuned are the number of estimators and the maximum depth of the trees. I used cross validation with 5 folds to find the best hyperparameters. The best hyperparameters are 50 estimators full depth trees. The best score is **0.96**

Feature importance To understand how the model uses the given features , a shap summary plot was used :

As the paper [2] suggests , the most important features are 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.

On the contrary, the least important features are the patient weight (in kg) and patient height (in cm) and the other indexes computed from these two features.

3.2.2 Support vector machine model

I also tried to use SVM to classify the data. I used a grid search to find the best hyperparameters for the model. The hyperparameters I tuned are the kernel and the C parameter. I used cross validation with 5 folds to find the best hyperparameters. I got as best score : 0.9099999999999999 best parameters : {'C' : 10, 'tol' : 1} . The score is less than the one obtaines with the RF classifier and It is to note that I used normalisation on the training data to avoid the effects of large values on the model. and since the SHAP values yealded made less sense . I used the RF classifier for the final submission.

Références

[1] 2D-3D Fully Convolutional Neural Networks for Cardiac MR Segmentation link

[2] Automatic Segmentation and Disease Classification Using Cardiac Cine MR Images : link

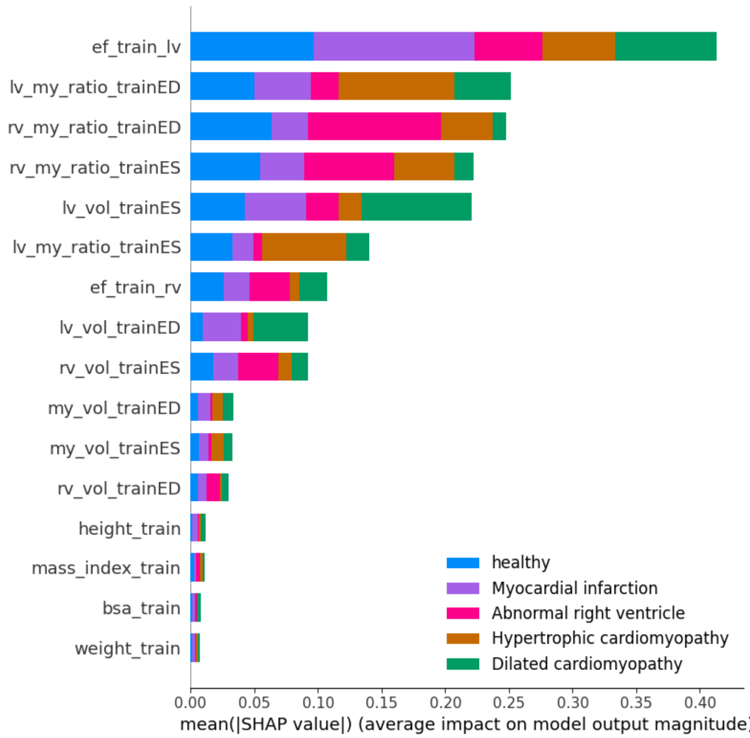


Figure 7 – Features ranked by their influence on the classifications