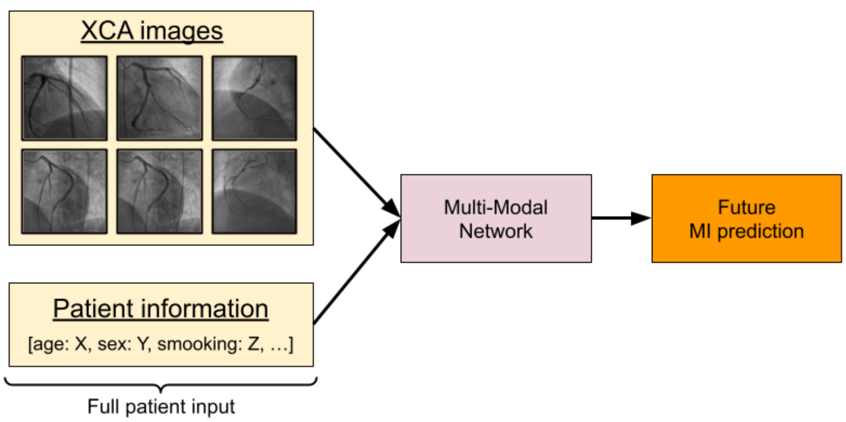
Laboratoire de Traitement des Signaux 4 (LTS4)

A multi-modal Deep Learning approach for Myocardial Infarction prediction

Master Thesis Report

Student: Ivan-Daniel Sievering
Supervisors: Dorina Thanou & Ortal Senouf
Professor: Pascal Frossard



École Polytechnique Fédérale de Lausanne

June 28, 2022

Contents

Abstract

Myocardial Infarction (MI) is a leading cause of death worldwide [1]. Its early prediction is thus crucial. In this work, we propose to predict if a patient may suffer a future MI from his/her X-ray Coronary Angiography images (CA) and his/her personal information, jointly and separately. In opposition to previous works, the prediction will be considered at the patient level and not at the artery sections'. In recent years, the number of fields in which Machine Learning is used is getting broader and broader. For Image Analysis, Deep Learning is becoming the golden standard, and its potential has already been assessed in medical applications, for example, in medical imagery [2].
In order to train such algorithms, a dataset gathered by the CHUV (Lausanne's University Hospital) has been used [3]. For each patient, his/her personal information and CA images have been recorded. The stenosis responsible for the MI has been annotated by the doctors and used as the ground truth during the networks' learning. The dataset is challenging: the amount of data is low (<500), there is a limited number of positive cases (<11%), and the artery sections have a heterogeneous size.
Three Deep Learning architectures have been proposed. The first one uses an Artificial Neural Network (ANN) to predict future MI only from patient information. The second is based on a Convolutional Neural Network (CNN) that processes CA images and patient information. The third approach extracts patches along the artery of the CA images and feeds them to a Transformer Network, along with patient information.
The ANN, despite using only the patient's information, achieved interesting performances (F1-Score: 0.30±0.09; AUC-ROC: 0.67±0.04; Precision: 0.20±0.08 & Recall: 0.70±0.11). The CNN reached the best F1-Score (F1-Score: 0.36±0.12; AUC-ROC: 0.67±0.04; Precision: 0.36±0.18 & Recall: 0.44±0.10). The Transformer showed an interesting recall despite a lower F1-Score (F1-Score: 0.27±0.04; AUC-ROC: 0.67±0.05; Precision: 0.17±0.03 & Recall: 0.63±0.13).
There is no literature with the exact same objective, and thus the performances cannot be assessed. The dataset is challenging to draw conclusions; more work is required to provide an implementable solution. But, our best networks reach a performance comparable to a interventional cardiologist on the same dataset, which motivates further investigation. There is plenty of room for improvement: other approaches and enhancements to tackle the different challenges will be proposed at the end of the thesis.

1. Introduction

1.1 Data-driven approach for MI prediction

Myocardial infarction (MI), or heart attack, is a leading cause of death worldwide [1]. Stenosis is one of its main causes. It is the shrinkage of a local segment of the artery which diminishes the amount of oxygen the heart receives. In case this reduction is significant, the stenosis may lead to a MI.

In this work, we propose to predict if a patient may suffer a MI in the following years (up to five). In our approach, we care about the consequence of the vessel's state and not only about the presence of the stenosis. Such early detection is based mainly on the physician's experience, and its automatisation could save lives. This objective differs from most previous works, which mainly focused on detecting (significant) stenosis.

For some years, the Image Analysis field has seen the emergence of Deep Learning algorithms. Such technology has been used successfully in medical applications like medical imagery [2]. Motivated by the success of this technology, it was decided to consider its potential for the future MI prediction task.

To handle this challenge, a dataset gathered by the CHUV (Lausanne's University Hospital) is used [3]. It contains two primary sources of information: a registry of patient information (age/sex/...) and a data bank of X-ray Coronary Angiography images (XCA), which contains for each patient two views of three different arteries (LAD/LCX/RCA). These images have been annotated by physicians who indicated the section which was the MI cause. This dataset is challenging for several reasons: it has a significant imbalance (<10% of the patients suffered MI), the number of patients is low (<500), the arteries' sections have heterogeneous sizes, and the images are big while having a small amount of information (most of the image contains background).

With this dataset, the previous Master Thesis [4] could not achieve good predictive performance by working at the artery section level. One hypothesis, along with the dataset's quality, was that the vessels are not rich enough to predict a MI event. To tackle this issue, the current Master Thesis proposes to work at the patient level instead, hoping that, at this level, the network would have enough information to predict a future MI and compensate for the data quality.

Different approaches have been considered to predict future MI at the patient level. First, the predictive power of the patient data has been assessed by considering different traditional Machine Learning algorithms (Random Forest, ANN, ...). Second, the predictive potential of XCA images has been analysed with Convolutional Neural Networks (CNN). Third, to profit from the artery's connectivity and remove the background in the input, patches have been extracted along the artery on XCA images. The resulting sequence has been fed to a Transformer network. Both the CNN and the Transformer were trained with and without access to the patients' information.

Despite the lack of similar literature to judge their performances, we can assess that our networks are competitive as they reach a comparable to the one of a interventional cardiologist on the same dataset. However, further work is needed before considering real-world applications. To reach this objective, interesting learnings, ways of improvement and new ideas will be proposed.

1.2 Related works

To the best of our knowledge, very few previous works tackle the challenge of predicting future MI from XCA with Deep Learning. We are unaware of recorded physicians' performances on future MI prediction from XCA (we only have the performance of one doctor on our testing dataset). Thus, it will be hard to assess the quality of our results. However, the literature is more extensive regarding detecting (significant) stenosis or predicting MI from other types of data than XCA. In what follows, we will first discuss the literature regarding future MI prediction from patient data. Then, related works that detect stenosis from Coronary Angiography images will be presented. Finally, some works that predict a future MI from Coronary images will be introduced.

1.2.1 Patient data

Some works focus on predicting MI from patient data. Some detect "close" MI (i.e. within less than a month). Such works achieve very high performances (i.e. F1-Score>0.75 & AUC>0.9). For example, a paper [5] proposes to use Decision Tree and Random Forest to predict MI in the following days. They reach an F1-Score of 0.97 and an AUC of 0.99. Another paper [6] uses Artificial Neural Network (ANN) technology to predict MI within 14 days. They reach an F1-Score of 0.78. A third paper uses Gradient Boosting [7] to predict MI within 30 days. They reach an F1-Score of 0.82 and an AUC of 0.96. The problem faced by these three papers is very different to the one in this report. In our work, patients are only considered *after* 30 days, and no features about symptoms or medical examination are available. Future prediction is a more complicated task. Thus, these works only confirm that ML methods using patient data can have predictive performance for MI events. Closer to our objective, a paper [8] compares Logistic Regression, Random Forest, Gradient Boosting and ANN to predict MI within six months. Their dataset is much bigger (2M patients), and they use all the features at their disposal (8k features). The best performance they achieve is an F1-Score of 0.101 (and AUC of 0.83) with a Shallow Neural Network.
The closest paper to our objective [9] predicts a patient's readmission one year after its first MI. However, they have more patients (7k) and use more features (192). They compare different methods: Logistic Regression, Naive Bayes, SVM, Random Forest, Gradient Boosting and ANN. Gradient boosting is their best-performing method (AUC of 0.72). In their paper, they conclude that "we found that ML methods do not improve discrimination when compared with previously reported approaches.".

1.2.2 Coronary Angiography images

1.2.2.1 Stenosis

Many papers predict stenosis from Coronary Angiography images (XCA or CCTA (Coronary Computed Tomography Angiography)). Despite not being the same objective as us, MI is closely related to stenosis and thus, exploring what has been achieved for stenosis is relevant.
A first paper [10] proposes to use Convolutional Neural Networks (CNN) on XCA images to predict stenosis on small patches of the image (32x32 pixels). They use artificial and real datasets (both balanced). Their backbone is a pretrained ResNet50 [11]. The network achieves an excellent performance (F1-Score: 0.91, precision: 0.89 & recall: 0.94), proving that Deep Learning can detect stenosis even on low-quality images and relatively small datasets.
A second one, [12], proposes a whole pipeline to analyse XCA images. It consists of four blocks. The image is first processed by a network (Xception-like [13]) that computes the projection angle, then the artery is detected through another network (also Xception-like). After that, the different objects in the artery (sections, stenosis, ...) are detected with a RetinaNet [14]. Finally, an Xception-like network provides a stenosis probability. Their algorithm reached an AUC-ROC of 0.862.

Other works use U-net-like [15] architectures as a backbone to detect stenosis from XCA. In [16], a U-Net-like structure was used together with a ResNet-like to predict stenosis, reaching an F1-Score of 0.917. In [17], they use a U-Net-like network and a Deep Neural Network conjointly to segment the network and predict the lesions types and locations. For stenosis, they reached an F1-Score of 0.829.
Transformer networks are also used for stenosis detection on CCTA, like in [18]. Their network takes as input a sequence of patches taken along the artery and feeds its features to a transformer network that outputs, for each patch, the stenosis prediction. This network will be discussed in detail ahead because it is the backbone of one of our networks. Their network achieved an F1-Score of 0.79.

1.2.2.2 Myocardial Infarction

For this work, an interventional cardiologist predicted MI from our testing XCA images. His performances (F1-Score: 0.095 & Recall: 0.4) show that the task is challenging even for a trained physician. This further supports the potential of the automatisation of this task. In [19], they propose a Deep Learning framework to measure the plaque volume from CCTA. The proposed network is a Convolutional LSTM [20] that receives sequentially the CCTA entries. Based on the measures achieved by their network, they provide a future MI prediction that achieves an AUC score of 0.70 and a recall of 0.659.
As previously mentioned, little literature exists regarding future MI prediction from XCA images. The first step towards future MI prediction was done in [21]. Their work predicted if a lesion may lead or not to a MI. To do so, they trained a CNN on patches centred on existing lesions (from XCA). Their network was based on a ResNet backbone. It achieved good performance (F1-Score: 0.571 & recall 0.667) and even outperformed experimented cardiologists (F1-Score: 0.348 & recall: 0.444).
The next steps were done in a Master Thesis [4] from the same laboratory. In this work, the objective was to predict future MI from XCA artery sections. The first approach they considered was to predict MI from the two views of an artery section with a CNN (ResNet-18 based). This approached reached: F1-Score: 0.18 ± 0.20, precision: 0.26 ± 0.31 & recall: 0.16 ± 0.21. They argue that the poor results are mainly due to the resizing of the artery section, which size changes along with patients.
Thus, the second approach considered the entire image as input to avoid resizing. An R-CNN-like [22] architecture has been used to simultaneously detect the different artery sections and make the stenosis prediction. Nevertheless, the performance was poor (due to the limited number of data and the annotation quality).
For the third approach, they decided to stick with the whole image input and provide a custom attention channel so that the network focuses on the selected artery section. The network uses a ResNet-18 backbone again. The performance is slightly better than previously: F1-Score: 0.22 ± 0.04, Precision: 0.25 ± 0.13 & Recall: 0.22 ± 0.10.
These results are still low compared to the performance achieved for stenosis detection or culprit lesion. Thus, to ensure that their approach was correct, they provide the performance of their first network on the stenosis detection task: F1-Score: 0.66 ± 0.06, precision: 1.0 ± 0.0 & recall: 0.49 ± 0.06. These performances are comparable to the other works, supporting that their approach was working and that future MI prediction is just a more challenging task than stenosis prediction. They argue that the limited number of positive MI cases, the data quality and the sections' heterogeneity may alter the input and thus prevent the network from learning the MI detection. Another possible cause they evoke is that the vessel may not contain enough relevant information to predict future MI.

2. Clinical study

The CHUV has gathered the data used in this study. The SPUM-ACS (Special Program University Medicine - Acute Coronary Syndromes) registry is a cohort of consecutive patients admitted with acute coronary syndromes (MI or unstable angina) to four university hospitals in Switzerland between 2009 and 2017. Further details of this registry have been reported previously [3]. For the present study, patients hospitalised to the CHUV with X-ray Coronary Angiography (XCA) images available for analysis will be included (n=709). The primary clinical endpoints evaluated after five years of follow-up were: (i) MI, (ii) other revascularisation, (iii) all-cause mortality, (iv) a composite of all three. XCA images from this data set have already been fully annotated by the CHUV cardiology team. Thus, the complete dataset consists of two main sources: a registry of patient information and a data bank of XCA images. The two sources will be considered both separately and jointly.

2.1 Patient data

The patient information file has 1387 columns for 984 lines (patients). The columns contain very diverse information, and only a small subset of them has been considered:

- Columns used for patient identification: SJID;
- Columns used to define the target class: MI events and MI events' date;
- Columns that are MI risk factors, selected after a discussion with one of the collaborators of the CHUV:
  - CVD risk factors: age, sex (boolean), hypertension, diabetes, hypercholesterolaemia, previous CVD (boolean), smoking habit (categorical) and BMI;
  - Clinical risk factors: left ventricular ejection fraction (LVEF), Killip class (categorical), cardiac arrest (boolean), Grace score, Kidney function.

From these columns, the target has been defined as the patient having a MI event at least 30 days after the previous MI incident; else, it would likely be a complication of the previous MI and not a new event. Missing data were replaced by the median in the case of continuous values and by the most frequent category in the case of categorical or boolean data. Of the 984 patients, only 78 have a valid MI event (7.93%). The distribution of the different features is presented in figure 2.1. Their (linear) correlation can been seen in figure 2.2.
After inspection, the LVEF and the Grace score features have a lot of missing values (21.75% and 8.33%). Moreover, they are poorly correlated to the MI event (-4.39% and 1.72%). The Grace score is highly correlated to other features (62.69%, 47.34%, ...). Thus, despite their medical relevance, it was decided to remove them.
The final patient dataset consists of a patient identifier, one target (MI event) and 11 features (sex, age, BMI, diabetes, smoking habit, hypertension, hypercholesterolaemia, previous CVD boolean, Killip class, previous cardiac arrest and the Kidney function). The dataset is provided normalised or not.

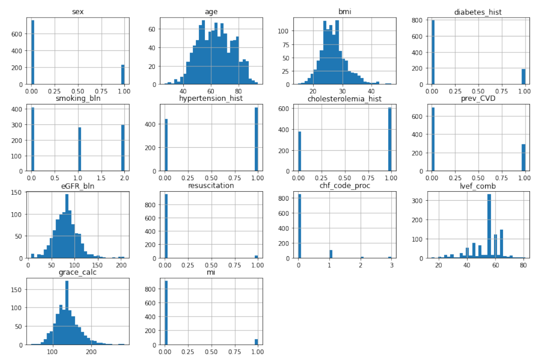


Figure 2.1: Histograms of the different features (and the target) contained in the patient data.

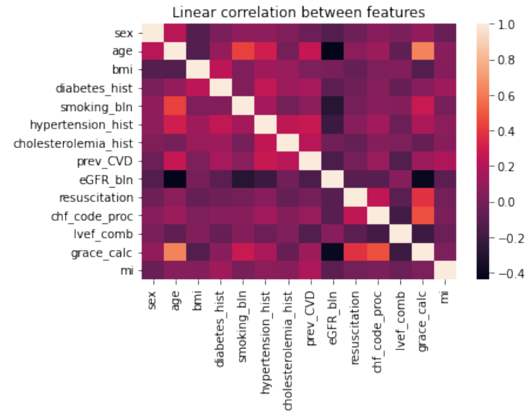


Figure 2.2: Correlation between the different features (and the target) contained in the patient data.

2.2 Coronary Angiography

The CA dataset contains 709 patients with six raw images for each: three different arteries (LAD, LCX and RCA) viewed from two different views (the angle varies from one patient to another). A typical raw image is presented in figure 2.3. Many patients miss images of an artery or/and a view. Others may have more than two views. Consequently, the total number of patients used later varies depending on the methods' needs and flexibility. In addition to each raw image, an annotated version of it is provided (example on figure 2.4). The physicians added boxes that subdivide the artery into segments. For each segment, a red dot was added inside the box if he/she considered the segment was the MI cause.

From the raw and annotated images, a dataset has been created. The dataset uses multi-indexing: patient, artery, view. Each row of the dataset (corresponding to a patient-artery-view combination) contains the path to the raw image, the coordinates of the boxes defining the sections and if the view includes a MI. As an indication, the complete dataset contains 48 patients with MI over the 709 (6.77%). In the patients with MI: 20 patients have only a MI in LAD, 12 only in LCX, 6 only in RCA, 2 in the three arteries and 7 in two arteries (3 in LAD+LCX, 3 in LAD+RCA and 1 in LCX+RCA).

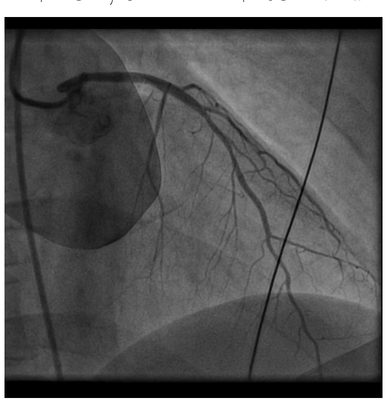


Figure 2.3: Example of raw CA image.

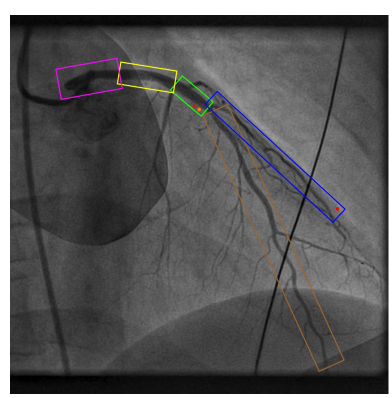


Figure 2.4: Example of annotated CA image.

2.3 Challenges

As already pointed out by the previous Master thesis [4], there are various challenges to face with this dataset:

1. Class imbalance;
2. Limited amount of data;
3. Boxes sizes heterogeneity;
4. Data quality;
5. Different source of data.

Class imbalance: The class imbalance varies depending on the method used, but the percentage of positive cases is never bigger than 11%, which makes the dataset highly imbalanced. Various balancing methods and specific losses designed for imbalance datasets have been considered.
Limited number of data: In practice, the number of patients is smaller than 500, whereas Deep Learning is data-greedy. The training dataset was extended to face this challenge. When available, pretrained networks were loaded to help early learning. Despite this, the number of data remains small, and some "sub-target" classes were always under-represented (for example, there are few examples of MI in RCA).
Boxes size heterogeneity: When the network is working at the section level, the disparity between the boxes size is challenging. Because the input dimension of a network has to be constant, the images were resized, and quality was lost. To reduce the degradation, the destination size depends on the box type. But, as shown in figure 2.5, the size of the boxes still varies a lot inside of a given section (figure from last Master thesis [4]).
Data quality: Despite the size of the images (1524x1524px), which is quite big for the Deep Learning field, the artery may be around 60 pixels wide at some points. Thus, there is a small information/size ratio. Additionally, to reduce the loading time caused by these images, a special data loading library has been used, FFCV [23].
Different source of data: The joint use of patient images and patient data is not trivial. Despite some literature, the best way to merge the features is not trivial, especially for specific networks (i.e. Transformers).

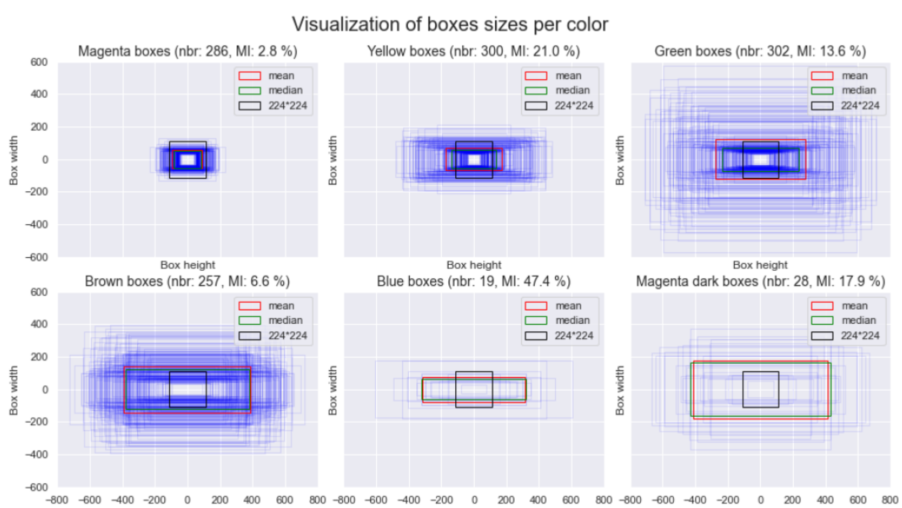


Figure 2.5: Heterogeneity of boxes' size between the different sections but also inside of a given section. **The figure from the last Master thesis [4].**

3. Problem definition

3.1 Objective

This thesis aims to predict future MI from a patient's information (age/sex/...) and his/her CA images. We follow a data-driven approach to tackle this challenge, inspired mainly by the recent advancements in Artificial Intelligence, particularly in Deep Learning. An illustrative image that summarises our learning framework is provided in figure 3.1. CA images and patient data undergo a data preprocessing block that prepares them for the next block, the Machine Learning (ML) algorithm. Then, this block outputs the future MI prediction. First, different ML algorithms that only use patient data will be tested. Next, two ML algorithms that deal with images are evaluated. Finally, the best ML methods for patient data are merged with the approaches for images.

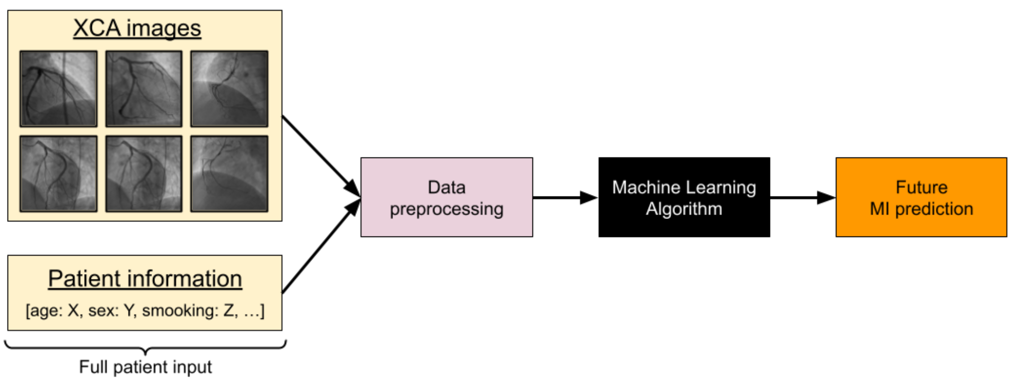


Figure 3.1: A learning framework for predicting future MI from XCA images and patient data.

3.2 General architecture

The precise nature of the input of the ML block (black block in figure 3.1) varies depending on the needs of the algorithm (full image, sequence of patches, ...). For each strategy, the preprocessing steps applied to the original data will be presented at the beginning of the related chapter.

However, the different ML algorithms receive a similar input structure: three pairs of two views (one pair for each artery) and patient information. Thus, we propose a common architecture. This architecture is presented in figure 3.2. It consists of four main blocks. Three blocks process each artery's CA images, and one block processes the patient information. From these blocks, using both the patient information and the CA images, a prediction at the patient level is provided.

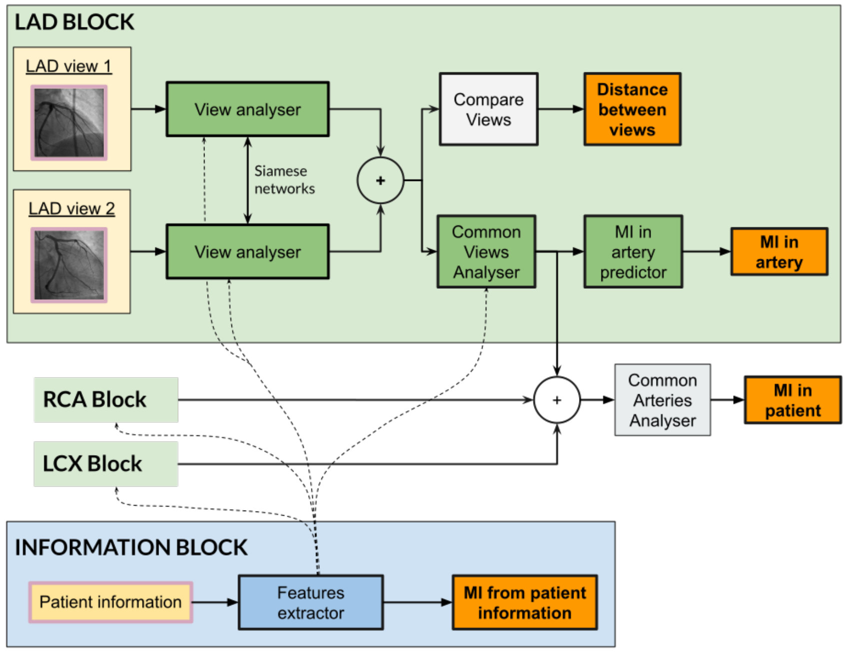


Figure 3.2: General architecture used by the Machine Learning algorithms (black block in figure 3.1). The architecture aims to predict future MI for a patient, given his/her basic information and his/her CA images. The input data (images and patient information) has already been preprocessed. The "plus" blocks symbolise concatenation operations. The dashed line provides features extracted from the "Features extractor" to other blocks.

The CA images come per pair (two views for each artery). The two views contain similar information as they describe the same object (the artery). Thus, the two views will be processed with siamese blocks (shared parameters). But, each artery will have its own siamese blocks. One may argue that, despite having common features, the views should be processed separately because each view may have different characteristics. There are several reasons the shared weights strategy is preferred. First, the angle between two views varies with patients, and thus specialisation would not be efficient. Second, the networks are big and sharing weights saves space. Third, after dataset inspection, the variance between the same view but from different patients is much higher than the variance between two different views from the same patient.

3.2.1 Artery block

Each artery block (in light green on the figure) receives the two preprocessed views as input. From them, it computes (i) a MI prediction at the artery level, (ii) a feature map that will be used at the patient level and (iii) a distance metric between the views. First, a "View analyser" block analyses each view. The output of these two blocks is concatenated into a bigger feature map that is processed through a "Common Views Analyser", which extracts information from both views. Its output is then used for two tasks: to participate in the patient-level prediction and to make the prediction at the artery level (through the "MI in artery predictor" block). Because the networks are siamese and the two views contain similar information, a distance metric can be computed between the feature maps extracted by the first and the second view. The "Compare Views" block does this computation and the score obtained is the "Distance between views". It is used to improve the learning procedure by providing the insight that the two inputs of the block (views) are related.

3.2.2 Common analysis

The output from each artery (output of the "Common Views Analyser") are concatenated. The new feature map is further analysed thanks to the "Common Arteries Analyser" to provide the final prediction at the patient level: "MI in patient".

3.2.3 Patient block

The information block (in blue in the figure) receives the preprocessed patient information. It goes through a "features extractor" block that extracts features and provides a MI prediction at the patient information level. Some features extracted by this block are provided to the arteries' blocks to contribute to the MI prediction at the patient level. Depending on the architecture, the features extracted from the patient information will go either to the "View analyser" or to the "Common Views Analyser".

3.3 Experimental settings

A similar method to train and evaluate all the proposed approaches is used in order to be consistent.

3.3.1 Evaluation metrics

As mentioned previously (section 2.3), the dataset is highly imbalanced. Thus, the accuracy (the percentage of correctly classified samples) is not a relevant metric as overfitting the majority class would lead to a good performance. Thus, it was decided to use the F1-Score [24], a metric derived from precision and recall. The F1-Score values are between 0 and 1, 1 being a perfect classification. These different metrics are defined as:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{3.1}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{3.2}$$

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.3}$$

Other metrics are considered, like the AUC-ROC (or just AUC), the Area Under the Curve of the Receiver Operating Curve. This metric is also between 0 and 1, 1 being the best performance. The curve considers different classification thresholds and computes the true-positive rate against the false-positive rate curve.

3.3.2 Training method

The dataset has been separated into training, validation and testing data (64%, 16% & 20%). The models have been trained using the k-fold cross-validation strategies. It consists of separating the training-validation dataset in k folds and training the network k times but changing each time the fold used for the validation (the other are used for training). The number of MI in each fold was enforced to be similar. Then, the mean of the performance on the k-folds is used as the model's performance; the same is done for the standard deviation. k varies depending on the time greediness of the model (5 for Neural Networks (NN), 10 else).
A model's hyperparameters (HP) were selected using the grid searches strategy. For the sake of space, the results of the grid searches are not displayed; only the best results are discussed. The dataset has been considered normalised and unnormalised when dealing with patient data. For the NN methods, the data images were not normalised. Different balancing strategies have been considered:

- Nothing (No): keep the raw data;
- Undersampling (Under): Remove non-MI cases until the number of MI and non-MI cases are identical;
- Oversampling (Over): repeating MI cases until the number of MI and non-MI are identical. Except for the Transformer and the ANN, where equality is enforced *per batch*.

3.3.3 Naive classifiers

In order to have milestones for comparison, the performance of our models will be compared to two naive strategies. The first naive approach always classifies the sample as negative (the dominant class). It will be referred to as *Always no-MI*. A second naive approach is a classifier that classifies the sample randomly as positive, following the positive distribution in the dataset. It will be referred to as *Random probabilistic*.

3.3.4 Losses

As displayed in figure 3.2, the networks will provide different predictions (MI in patient, MI in artery, ...). For the methods that require a loss function, different classification losses have been considered; they are further detailed in appendix A.1.1:

- Binary Cross-Entropy Loss (BCE);
- Focal loss [14]: a loss specialised for imbalanced datasets by reducing the contribution of easy samples. It has been applied successfully to the stenosis detection task [12];
- AUC loss [25]: a loss that optimises the AUC-ROC metric instead of the accuracy and is thus more suited for imbalanced datasets. More, there are strong ties between AUC-ROC and F1-Score. This loss has to be optimised with PESG [26]. It has been used to classify Mammograms [27].

The networks also compute the distance between the features map of each view. To do so, a distance loss is used. Such loss enforces the network to provide a similar embedding to two feature maps; in this case, the extracted features maps from the two views of the same artery (as one may assume both views should contain similar information). It is computed as the mean of the square of the pairwise euclidean distance between the two feature spaces. It is the "Compare Views" block from the general architecture 3.2.
These losses are combined through a weighted sum whose weights are HP. The principal loss is always the prediction at the patient level and thus has its weight fixed to 1. The formula is given just below:

$$loss = loss_{patient} + \sum_{i=1}^{i=n} w_i * l_i \tag{3.4}$$

Where w_i is the weight (between 0 and 1) of the auxiliary loss l_i and n the number of auxiliary losses.

3.3.5 Interpretability

For the methods that use CNN and the Transformer, interpretability methods have been applied to their best-performing network. Interpretability consists of understanding what the network learned. From this analysis, a human can understand if the network has the expected behaviour. Some methods show the importance of some parts of the image for the network through gradient computation: Saliency map [28], Integrated Gradient [29] and GradCAM [30]. Input occlusion [31] shows the importance of the regions of the image by hiding parts of it. Finally, FGSM [32] generates adversarial images to challenge the robustness of the model. The methods and their implementation are further introduced in appendix A.1.2.

4. MI prediction: a traditional Machine Learning approach

The first approach has been to evaluate the predictive performance that could be achieved using only patient data. In this chapter, only the "Information Block" of the general architecture (blue block in figure 3.2) is considered, and its prediction is considered as the prediction at the patient level. The patient information dataset of section 2.1 has been used without more processing.

4.1 Methods

4.1.1 Classic Machine Learning

Different Machine Learning (ML) classifiers have been considered, they all have been implemented using *Sklearn*. Because they are very well known, they won't be introduced in this report.

- Decision Tree [33];
- Random Forest [34];
- Balanced Random Forest [34];
- Support Vector Machine (SVM) [35];
- Logistic Regression [36];
- Naive Bayes [37];
- Gradient Boosting [38].

4.1.2 Artificial Neural Network

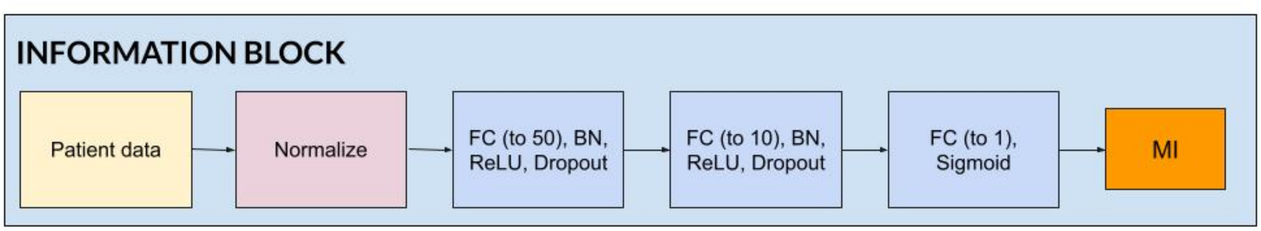


Figure 4.1: Proposed ANN structure to predict future MI from patient data. Possible implementation of the Information Block of the general architecture 3.2.

Inspired by a paper that uses clinical data to predict MI with an Artificial Neural Network (ANN) [6], an ANN has also been implemented. However, knowing that the ANN would be implemented in a more extensive Deep Neural Network (DNN) framework, it was decided to use the architecture from [39] which presents a way to merge patient information with Convolutional Neural Networks (CNN). Both papers propose similar architectures. Our network takes all the information as input and outputs the future MI probability. A hidden layer of this network consists of an FC layer, batch normalisation (BN), ReLU and dropout. The classification layer is an FC layer and a Sigmoid activation.
Two extensive grid searches using SGD optimiser on BCE loss indicated that the best architecture consists of two hidden layers, one with 50 neurons and then one with 10 neurons. The architecture is illustrated in figure 4.1. The best initialisation follows a Kaiming normal distribution [40].
Once the architecture defined, the other HP were further inspected, and other losses were considered: BCE (optimised by SGD), Focal (also SGD), AUC (optimised by PESG) and AUC with BCE pre-training. The networks were trained during 300 epochs, with a batch size of 32 and a scheduler dividing by 10 the learning rate after 25 epochs without improvement. For the AUC with BCE pre-training, the network was first trained with SGD optimiser on BCE for 200 epochs and then with PESG optimiser with AUC loss for 300 epochs.

4.2 Results

Table 4.1: Performance of the different methods to predict MI from patient data.

Model	F1-Score (mean±std)	Accuracy (%) (mean±std)	Normalise	Balancing	Stratify
Always no-MI	0.00	**92.07**	-	-	-
Random probabilistic	0.08	85.40	-	-	-
Decision Tree	0.23 ± 0.08	70.18 ± 0.08	Yes	Undersample	No
Random Forest	0.16 ± 0.12	60.06 ± 29.4	Yes	Undersample	No
Balanced Random Forest	0.21 ± 0.11	66.23 ± 19.97	Yes	Undersample	Yes
Support Vector Machine	0.15 ± 0.05	17.28 ± 3.59	Yes	No	Yes
Logistic Regression	0.24 ± 0.10	68.37 ± 5.03	No	Oversample	Yes
Naive Bayes	0.24 ± 0.09	65.63 ± 5.03	No	Oversample	Yes
Gradient Boosting	0.24 ± 0.06	71.36 ± 2.95	Yes	Undersample	No
Artificial Neural Network	**0.30 ± 0.09**	81.22 ± 8.04	Yes	Oversample	No

Table 4.2: Performances and HP of the ANN to predict MI from patient data with different learning strategies.

Training strategy (optimizer+loss)	F1-Score (mean±std)	Learning rate	Weight decay	Dropout (%)	SGD momentum	Others
SGD+BCE	0.25±0.05	0.055	0.00175	48.36	0.2674	-
SGD+Focal	**0.31±0.03**	0.054	0.0066	47.9	0.96	Focal alpha: 0.48 Focal gamma: 0 Focal reduction: sum
PESG+AUC	0.28±0.06	0.04	0.0048	49.76	-	PESG margin: 0.92 PESG gamma: 470
SGD+BCE into PESG+AUC	0.30±0.09	0.0084 0.0636	0.0023	46.98	0.265	PESG margin: 0.82 PESG gamma: 495

4.3 Discussion

4.3.1 ANN training strategies

From Table 4.2, the performance of the different ANNs can be compared. As expected, losses developed for imbalanced datasets (Focal, AUC) outperform the BCE. The network using AUC with BCE pre-training outperforms the ones using only AUC. The authors of the loss already described this behaviour. The model using the Focal loss obtains the best performance (F1-Score of 0.31) while having the lowest standard deviation (0.03). In the following, despite the Focal loss being the best for this network, it will be considered with BCE or AUC loss because the image analysis networks are not using Focal loss. In appendix A.2.1, the evolution of the F1-Score along epochs for each strategies is discussed. The main outcome is that one can clearly see the benefit of switching from BCE to AUC loss.

4.3.2 Comparison between methods

Table 4.1 shows that all the proposed methods outperform the naive ones in terms of F1-Score, indicating that they could extract knowledge from the data. Despite using very different technologies, the methods obtain comparable performance. That may indicate that the maximum predictive capacity of the patient features may have been reached (or a difficulty gap). Random Forest performs poorly, but its counterpart (Balanced Random Forest) achieves good performance. The poor performance of the SVM has no clear explanation as there is no conceptual motivation for it to perform poorly than the others. ANN outperforms the others but has significantly more parameters. In appendix A.2.2, the impact of the different dataset extensions are discussed. The method that showed the best performance is ANN. It achieves the best validation F1-Score (0.30). It also has higher accuracy. It reaches (on the validation set) an AUC of 0.67 $\pm$0.04; a recall of 0.70 $\pm$0.11, and a precision of 0.20 $\pm$0.08. Moreover, the ANN is more easily implementable in a DNN framework than the other approaches.

4.3.3 Comparison with previous works

In the Related Works section (chapter 1.2.1), different papers that use patient data to predict future MI have been presented. The ones that predict MI in less than a month can not be compared to the performance achieved above because the task is different.
In [8], they predict MI within six months, which is already closer to our objective. As a reminder, they achieve an F1-Score of 0.101 (and AUC of 0.83). Our performance outperforms their F1-Score and reaches a bit lower AUC while using much less data (regarding the number of patients and features). They also compared the various method that we presented and also show that no method completely outperforms the others, except the Neural Network (NN), as for us.
The paper closest to our objective [9] predicts patient readmission one year after his/her first MI. They have much more data at their disposal (7k patients). Again, the performances of their different methods are very similar (except for the naive Bayes). Their best method is Gradient Boosting (AUC of 0.72), our second best method.
Our results are coherent with all these papers: comparable (and even better) performance is reached with fewer data. So, choosing fewer but smartly selected features is the correct approach. Because the performance of all the comparable papers and our results are similar, it raises the question again: "Is the maximum predictive capacity of the patient data reached?" (or at least a considerable difficulty gap).

5. MI prediction: a CNN approach

Many Image Processing tasks can be viewed as filters applied to images. One way to describe this operation mathematically is by using 2D-Convolutions of a kernel on an image. That is precisely what is done by Convolutional Neural Networks (CNN). They learn which kernel applies to extract the needed features. CNN has been used successfully for image analysis for almost ten years. Thus, it was decided to consider a CNN for the CA image analysis in the first place.

5.1 Dataset preprocessing

The dataset used here is based on the dataframe presented in chapter 2.2. Because we are working at the patient level and CNNs are not flexible to missing data, only the patient with a complete input can be considered. Thus, only the patients having at least one view of each artery are selected. If the patient has only one view of the artery, the view is repeated and considered the "second" view. Of the 709 patients, only 445 remain after this selection; along with them, 47 have a MI (10.6%) (just one MI has been lost). In the reduced dataset, each patient has six images (two views of the three arteries). These images do not contain the boxes from the physician, and this information is valuable. To give this insight, a mask layer was added to each view of each artery. This mask layer consists of a greyscale image on which a multivariate Gaussian has been drawn for each annotated section of the artery (box). Figure 5.1 illustrates how the mask is created, from the annotated image. Figure 5.2 shows a complete input for the CNN.
The dataset has been considered with different balancing strategies. The dataset preprocessing steps and extensions are presented in appendix A.3.1.1; they mainly aim to tackle the small number of data by generating "altered" samples from the existing ones. Due to the size of the dataset, it had to be loaded using a specific library, FFCV [23]. The dataset's large size and the library's performance are discussed in appendix A.3.1.2.

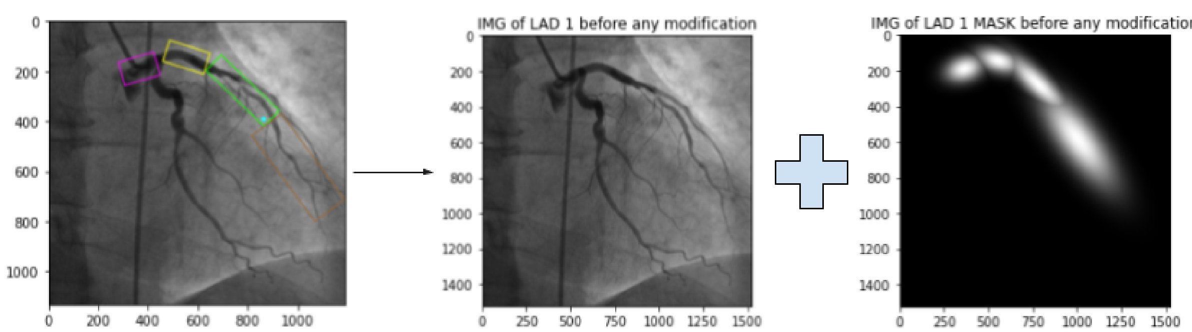


Figure 5.1: Example of how an annotated image is converted to the raw image and its mask, where each box is converted in a multivariate Gaussian distribution.

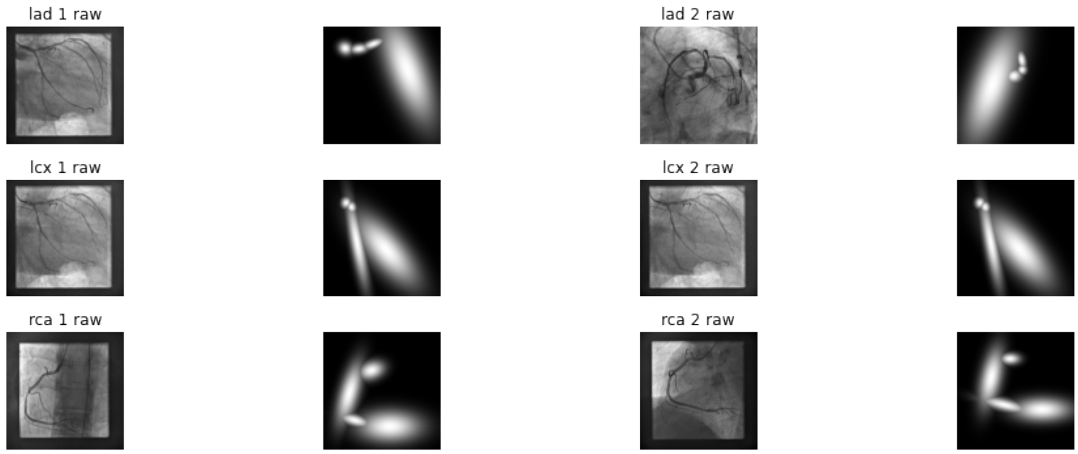


Figure 5.2: Example of a complete input to a CNN architecture. Each patient has images of the two views of the three arteries along with their masks based on the physicians' boxes.

5.2 Methods

5.2.1 CNN with arteries features common analysis

In the first place, the potential of CNN has been considered without patient information. The CNN have been implemented inside of the general architecture illustrated in figure 3.2. The resulting network, presented in figure 5.3, is thus very similar to the general architecture. Each view and its mask are processed by a ResNet-18 [11], which is a known and efficient CNN and has been used by other works using CA to predict future MI [4] [21]. The ResNets have been modified to take two channels as input (instead of three), the two channels being the raw image and its mask. Its three last layers (classification ones) have been removed. The concatenation of the features extracted from the two ResNets is not processed jointly and is directly sent to the patient-level analysis block and to the artery predictor.
At the artery level, average pooling is applied to the concatenation of the features extracted from the two views. Then, the data is flattened, and dropout is applied. The resulting vector undergoes a Fully Connected (FC) classification layer activated by sigmoid, resulting in the prediction at the artery level.
The features extracted from the three arteries are concatenated at the global scope. Next, they are processed in three steps. First, the dimension of the data is reduced by max pooling. Second, a residual convolutional block is applied (two convolutional layers with BN and ReLU activation, connected by a residual connection). Third, the feature map is converted to a vector by average pooling, flattening, and dropout. A classification FC layer activated by sigmoid then makes a prediction at the patient level.

5.2.2 CNN with maximum of arteries prediction

The second approach is a slight modification of the previous one. It has been motivated by the fact that MI is a local event; thus, the common analysis of the arteries' features should not add information (i.e. the prediction of if an artery may lead to MI is not improved by information from other arteries). Thus, the idea was to remove the "Common Arteries Analyser" from the previous network and change it by a max function between the prediction of each artery. The result is presented in figure 5.4. Note that this network uses more than five times fewer parameters than the previous one.

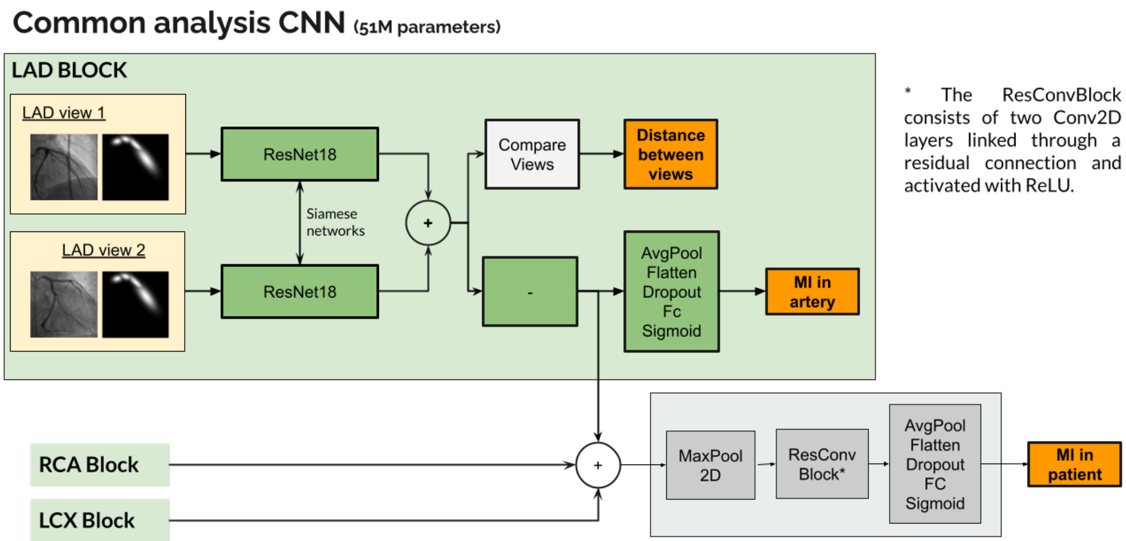


Figure 5.3: Architecture of the CNN common network. The network aims to predict future MI for a patient, given his/her CA images. The images are processed by CNNs. The "plus" blocks symbolise concatenation operations. The "-" inside of a block indicates that it does nothing to the data.

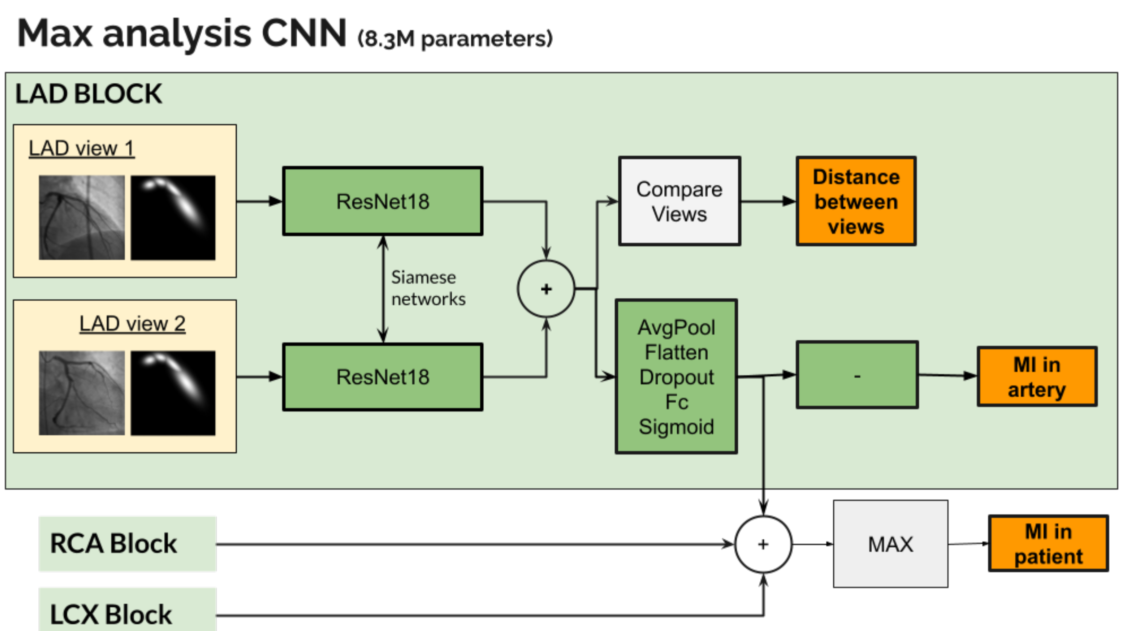


Figure 5.4: Architecture of the CNN max network. The network aims to predict future MI for a patient, given his/her CA images. The images are processed by CNNs. The architecture profits from the MI being a local event by just considering the maximum prediction of the three arteries. The "plus" blocks symbolise concatenation operations. The "-" inside of a block indicates that it does nothing to the data.

5.2.3 CNN with patient data

Once the performance of the CNN with the max architecture was assessed, it was decided to evaluate how the patient information could improve its performance. To do so, the patient data analysis network implemented in section 4.1.2 is used as "Information Block". The 10 features extracted by this network after its second FC layer are concatenated to the features extracted by the CNN before the artery level prediction, as shown in figure 5.5.
Different approaches to profit from the images and the patient information could be considered. Various paper propose the concatenation approach that is used here: for plankton images classification ([41]), to improve the detection of skin diseases from images and patient information ([42] & [39]) and to predict cardiovascular disease from myocardial perfusion images and patient information ([43]).

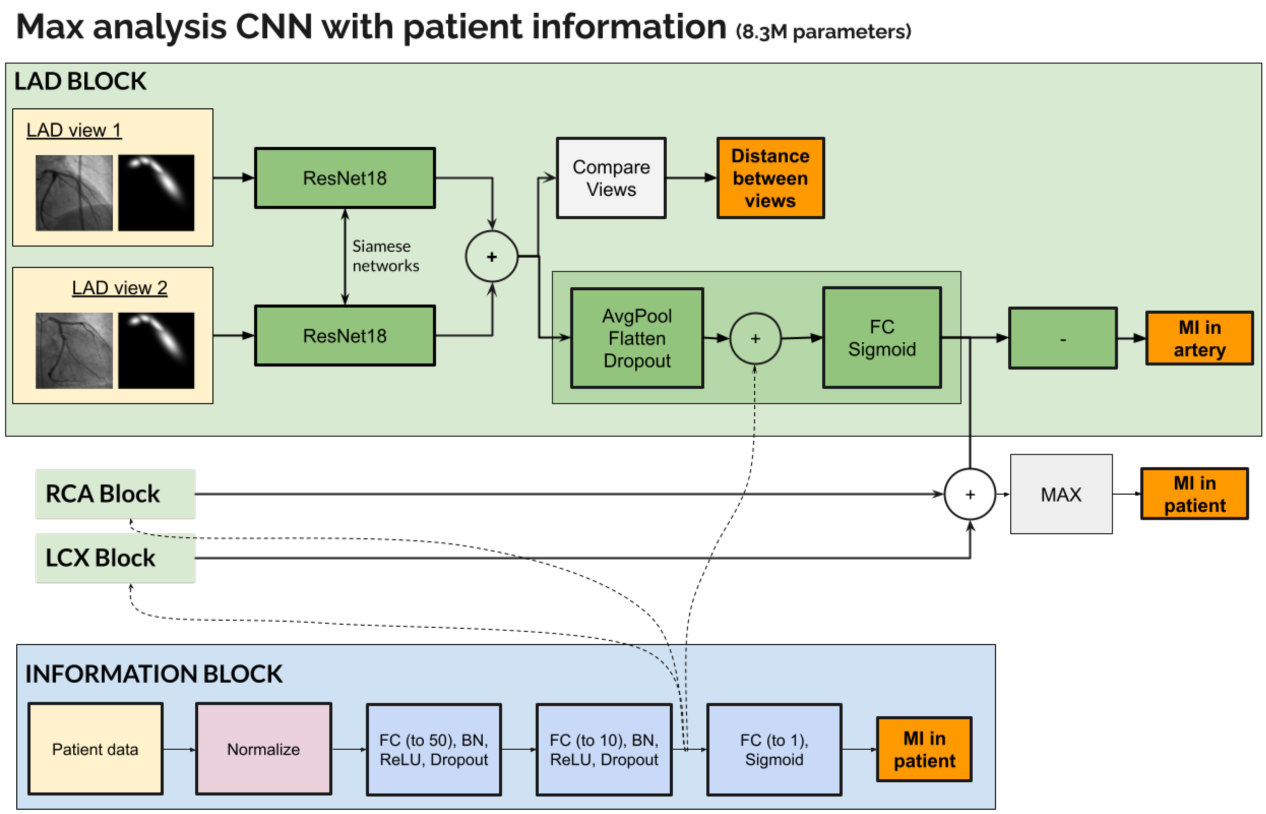


Figure 5.5: Architecture of the CNN max network with patient data. The network aims to predict future MI for a patient, given his/her basic information and his/her CA images. The images are processed by CNNs and an ANN (from figure 4.1). The "plus" blocks symbolise concatenation operations. The "-" inside of a block indicates that it does nothing to the data.

5.3 Results

5.3.1 Configuration

The networks have always been trained during 20 epochs with a batch size of 4. A scheduler divides the learning rate by 10 if no improvement is detected after 3 epochs. The weights are initialised with a Xavier uniform distribution [44]. The training duration of a CNN was quite long due to the issues mentioned previously. Depending on the architecture, a 5-fold validation may last between eight and twelve hours. This significant duration implies that the grid searches could only explore a small subset of the hyperparameters' space.
The Max network with patient data has been considered both with and without pretrained networks: pretrained ANN (best performing ANN on patient information) and pretrained CNN (best performing CNN only on patient images). It has also been trained with a higher weight on the arteries' losses to see if it could help the network using all the arteries. In section A.1.3, the list of the HP considered during the grid search is provided. As a reminder, the weights between the global MI prediction loss and the auxiliary losses are HP.

5.3.2 Performance on the clinical data

The performances obtained by the different configurations are presented in table 5.1. The first architecture is referred to as "Common", the second one as "Max", and the third as "Max with patient data". The three architectures have been evaluated in different configurations of balancing method and loss function. The F1-Score is computed at the patient and artery level. The F1-Score at the artery level is the prediction done inside an artery block. These values have to be handled with care for several reasons. First, the prediction done at this step may not be used afterwards for some networks. Second, some networks had small weights on the arteries' prediction losses. Third, the number of MI for each artery is small. There is much more MI in LAD than in LCX and in LCX than in RCA. Plus 5-fold cross-validation is used, which reduces further the number of MI. A network may be trained only with one or two examples of MI in a specific artery during a fold. So, the results related to these values must be considered with care. The evolution of the training and testing F1-Score along epochs is provided in appendix A.3.2.1 for the different networks.

Table 5.1: Performance of the CNN architectures to predict MI from CA (and patient information). The "Common" architecture is presented in section 5.2.1, the "Max" in 5.2.2 and the "Max with patient data" in 5.2.3.

CNN Model	Balance	Loss	Other	F (valid) (mean ± std)	Artery F valid (mean) (LAD/LCX/RCA)	AUC-ROC (valid) (mean ± std)
Always no-MI	-	-	-	0.00	0.00 / 0.00 / 0.00	0.50
Random probabilistic	-	-	-	0.1056	0.06 / 0.04 / 0.03	0.50
Common	Over	BCE	-	0.31 ± 0.07	0.00 / 0.00 /0.00	0.65 ± 0.03
	Over	AUC	-	0.30 ± 0.12	0.00 / 0.00 / 0.00	0.66 ± 0.07
Max	Over	BCE	-	0.28 ± 0.09	0.00 / 0.17 / 0.00	0.63 ± 0.06
	No	AUC	-	0.26 ± 0.05	0.00 / 0.06 / 0.00	0.63 ± 0.08
	Over	AUC	-	0.30 ± 0.11	0.00 / 0.14 / 0.00	**0.67 ± 0.04**
	Over	Focal	-	0.28 ± 0.08	0.05 / 0.09 / 0.03	0.65 ± 0.06
Max with patient data	Over	AUC	-	**0.36 ± 0.12**	0.11 / **0.28** / 0.00	**0.67 ± 0.04**
	Over	AUC	Big weight for artery loss	0.27 ± 0.16	**0.33** / 0.06 / **0.06**	0.60 ± 0.09
	Over	AUC	Pretrained	0.35 ± 0.18	0.02 / 0.30 / 0.00	0.65 ± 0.04

5.4 Discussion

5.4.1 Common networks

The common network reaches almost the same performance when trained with BCE or AUC loss; the variance is even lower with the BCE loss than the AUC. In both cases, the network cannot learn to predict MI at the artery level. For this network, the prediction at the artery level is not essential as the features are further processed. The grid searches show that (for both losses), the network reaches a higher global F1-Score with a small weight on the prediction at the artery level. Thus, these additional losses may not be beneficial and complicate the learning. Moreover, as mentioned in the Results section 5.3.2, these values must be considered with care due to the small number of MI in each artery.

5.4.2 Max networks

In opposition to the common network, the AUC loss with overbalanced dataset outperforms the BCE loss when used on Max network. This time, the AUC loss helps the network to learn a better classification; the AUC loss being closer related to the F1-Score than the BCE loss. Balancing the dataset when using AUC still helps. The Focal loss performs similarly to the BCE, which is not enough to explore further as it adds more HP. Regarding the F1-Score at the artery level, most configurations can learn how to classify LCX without outstanding performance. The Focal loss is the only one able to take a bit of profit from the three arteries. Again, these values have to be handled with care.

5.4.3 Max with patient data

The vanilla Max with patient data network reaches a higher performance than the one with a higher ratio on the arteries; a high focus on these additional objectives complicates the learning process. The vanilla network reaches a comparable performance to the one that preloaded modules.
Regarding the F1-Score at the artery level, the vanilla uses both LAD and LCX, whereas the one with a higher weight on the artery uses the three, meaning that the additional loss achieves its objective. Interestingly, the network with a higher weight on arteries uses more the LAD than the LCX, in opposition to all the others. That makes sense as most of the MI cases are in LAD.
When looking at the evolution of the performance of the preloaded network (figures A.9 and A.10), there is no clear benefit of the preloaded modules: the performance in validation starts at the same point than the ones with random initialisation. Despite starting with a better F1-Score in training, the F1-Score then vanishes, and "new" learning starts. The network may have difficulty learning the new joint classification layer and adapting to new data.

5.4.4 Architectures comparison

The results show that all the proposed methods achieve a better patient performance than the naive implementations. However, some cannot outperform the random one at the artery level. The best performance is obtained by the Max network with patient data, indicating that the network can extract some knowledge from the patient information on top of the CA images (it also outperforms the prediction only from the patient information, see table 4.2). Globally, the max and the common architecture achieve similar performances, but the first one is much lighter and makes more sense from a medical point of view.
Interestingly, most networks learned MI in the LCX artery and not in the LAD, which has more cases. We may assume that MI in LCX is more accessible to detect than in LAD for the networks. MI in RCA is barely detected; this may be due to the significantly small number of MI in this artery. No network can reach an excellent predictive score at the artery level, but the Max architecture is more promising in this sense.

5.4.5 Interpretability

Different interpretability methods will be discussed. They have been introduced in section 3.3.5 and will be applied to our best performing CNN, Max with patient information (figure 5.5). From these results, it is impossible to assert that the network learned how to use the input correctly, but there are some interesting behaviours, although very artery dependent.

Inspection A positive input's interpretability is shown in figure 5.6. The red rectangles indicate sections that contain MI. As a reminder, the input is the full images and their mask, but the interpretability analysis is done at the artery level. For the GradCam, the given attention heatmap is common for the two views.
The Saliency maps of LAD and RCA are similar to the mask and do not have additional details. For LCX, the non-negligible values are mainly in the mask. However, some greater attention is given to other points. For the first view of LCX, the network looks at the beginning of the artery, which contains some artefacts. For the second view, the network focuses on an area in a red rectangle (MI's area).
The integrated gradient is not meaningful for LAD and RCA (except for the second view of LAD that draws the mask's shape). For LCX, the Integrated Gradient highlights the same information as Saliency maps.
The result on LAD for GradCam is surprising; it only focuses on minor points. However, two of these three points are in a MI rectangle for the first view. For LCX, the attention is focused on the arteries. The more intense area of focus is again the beginning of the artery but is a bit wider. Some attention is again given to the artefact. For RCA, the results make no sense; the network is just looking at the image's border.
For LAD and LCX, the main impact of occlusion is on the arteries. A void square on the artery increases the probability prediction of MI, whereas a void square on the background has a lower impact (a bit toward non-MI). The network gives thus more significance to the arteries than the background. The occlusions images make no sense for RCA
The adversarial input of LAD is very close to the input. However, it leads the network to change its prediction to 13.9%, showing that it is sensitive to adversarial attacks. On the LCX, the network is more robust as the image had to be deteriorated significantly to change the prediction. Finally, RCA is incoherent.

Analysis Different points emerge from these analyses. First, the network looks to the arteries more than to the background, which may be explained by real learning of what an artery is or simply utilising the provided mask. For LAD and RCA, it looks like the network is just using the mask. LCX may detect some artery-likeness features, although they are sensitive to artefacts.
Second, the network is much more activated, robust and performing on LCX. That was already suggested by the performance achieved (F1-Score of 0.11 on LAD, 0.28 on LCX and 0.00 on RCA) and is confirmed here. The results on RCA make no sense; the network may not use this artery due to the few numbers of MI in this artery. The max architecture may not help the network to learn distributively.
Third, the network seems to have a higher focus on MI-related areas. That is not obvious and does not mean that it will be able to make the correct prediction from it, but it indicates at least that it may have got a form of "intuition" behind the objective.
Finally, please note that the result displayed here is a "favourable" one and that some other interpretability analyses on other patients were less meaningful. The analysis of a negative input is very similar (an example is available in appendix A.3.2.2). The only substantial difference is that LAD is more robust to adversarial attacks, which may indicate that it is easier to trick the network from MI to non-MI than from non-MI to MI.

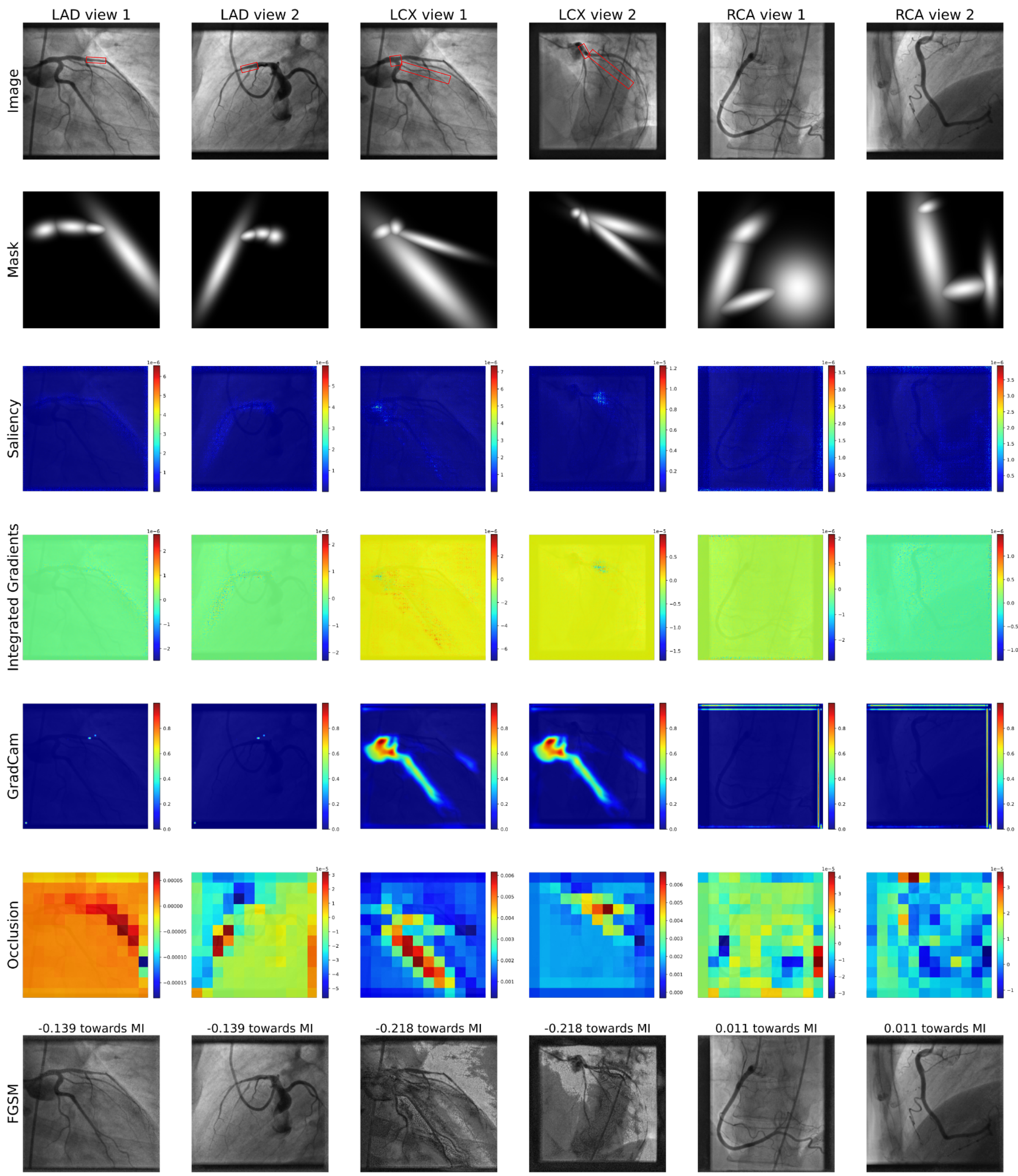


Figure 5.6: Interpretability analysis of the Max CNN with patient data architecture 5.5. The input contains MI inside of the red rectangles and comes from the testing dataset. The different interpretability methods have been introduced in section 3.3.5. The analysis is done artery wise. GradCam returns the same mask for both views. The impact on the prediction of the FGSM generated images is indicated in their title.

6. MI prediction: a Transformer approach

In the CNN architecture, most of the input images consist of a background that does not contain relevant information, despite an insight of the artery position given through the mask layers. Moreover, the input does not use the arteries' "connectivity" (i.e. the blood flow). The next idea was thus to extract patches along the artery and treat them as a sequence.
Various Deep Learning architectures deal with sequences (Recurrent Neural Network [46], LSTM [47], ...). Among them, the Transformer is promising; such a network learns how to interconnect the elements of the sequence to provide the best output. Thus, our sequence of patches is fed to a Transformer network, which will be used to predict the MI state. This approach is further motivated by a paper which uses successfully 3D patches of CCTA imagery to predict the stenosis along arteries [18].

6.1 Dataset preprocessing

From the main dataset (section 2.2), a new one has been created that emphasises the importance of the boxes, needed for the future patch extraction. The dark magenta boxes have not been considered (they indicate CABG). The blue boxes have also been removed because too few of them are available. These operations remove 2 MI from the datasets (and zero patient).
If we only consider the patients with all the information (the ones with the four boxes in the two views of the three arteries), only 344 remain, of which 19 with MI (the initial dataframe has 709 patients and 48 MI). Thus, another approach has been considered: accept the patient as long as they have at least one complete artery (four boxes on the two views). With this new method, the dataset has 466 patients, 40 MI (8.58%). However, 106 of them only have two complete arteries and 17 only one. A lot of MI are contained in these "partially complete" patients.
From this new dataset, patches are extracted along the artery's centerline, following the blood flow. The patches extraction is done by the next steps and is illustrated in figure 6.1. A lot more details about each step and its challenges are provided in the appendix A.4.1.

1. Extract each box (section);
2. Orient the box such that blood enters from the left and exits at the right;
3. Compute the centerline of the artery in the box. The detection is based on [45] and uses classical Computer Vision tools;
4. Apply preprocessing and data extension to the box and its mask (rotation, ...);
5. Extract semi-uniformly patches on the centerline from left to right (blood entrance to blood exit);
6. Create a list of patches for the whole image (the first patch of the global list is the first patch of the first section and the last patch of the global list is the last patch of the last section).

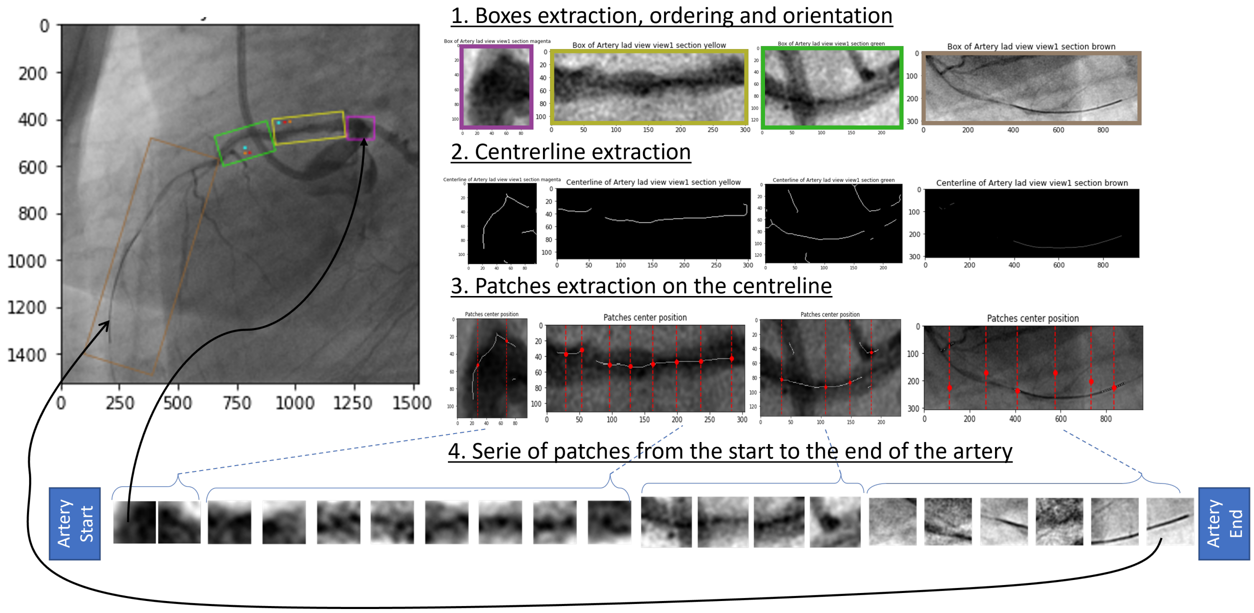


Figure 6.1: Simplistic scheme of the patch extraction strategy. The boxes are extracted, and a centerline is detected. Then patches are extracted from left to right along the centerline. The result is a list of patches that starts at the beginning of the artery and ends at its end. A complete input consists of six of these lists (two views for each one of the three arteries).

The size of the extracted patches has been fixed to 64x64px. The arteries are typically 60px wide; thus, this size is insufficient. A bigger size would have been interesting to ensure containing the whole artery width on the image, but this parameter significantly influences the network size. The number of patches extracted depends on the box type, as some tend to be bigger than others (figure 2.5). It has been decided to extract 32 patches for the magenta boxes, 64 for the yellow ones and 128 for the green and brown ones.

All the patches have been extracted sequentially (the first one on the list is the one from the left and the last from the right). The list of each section is concatenated following their apparition order (magenta-yellow-green-brown). That results in a sequence of patches from the start to the end of the artery. Finally, each patient has a sequence of 352 patches ($= 32 + 64 + 128 + 128$) with the blood flow logic for each view of each artery. The dataset can be raw, undersampled or oversampled (oversampling in batch).

The resulting sequences suffer from various defects. First, the rotation can be incorrect in some scenarios. Second, the centerline detection algorithm used is simple, but the quality of its output is variable. Other methods could have been considered, like U-Net, for example, [16]. However, their implementation would have required an annotated dataset. Third, the hypothesis that the blood outputs on the right may be wrong in the case of a "curvy" artery. Fourth, when there is a bifurcation (two sections emerge from one), they are considered in *serie* in the sequence despite being in *parallel*. A graphical representation of the sequence may be a solution to this issue.

6.2 Methods

A sequence of data can be analysed in different ways with DNN. In this chapter, a Transformer structure is used (it learns how to interconnect the different elements of the sequence). The backbone of this network is a transformer inspired by a previous paper TR-Net [18]. First, this network will be introduced. Then, it will be implemented inside of our general architecture without patient information. Finally, two different ways to merge patient information with the TR-Net will be proposed.

6.2.1 TR-Net

The transformer network used in this chapter is inspired from [18]. This paper predicts stenosis in the artery from CCTA patches taken around the centerline. The data and the problem being similar, it was natural to consider adapting it for our task. However, they deal with a more manageable task (stenosis is more straightforward to detect than future MI) while using richer data (CCTA images are 3D and have an annotated centerline). They can also use a more precise loss: they have the stenosis ground truth for each patch (and we only have the MI state at the section level).
Their network is called TR-Net (TRansformer-Network) and is divided into two main parts: a 3D-CNN (feature extraction) and the Transformer itself that interconnects the patches' features. The global architecture of TR-Net is presented in figure 6.2. First, each patch goes through the 3D-CNN. The layers composing the CNN are presented on the left of Figure 6.3; it consists mainly of convolutional layers activated with ReLU and max pooling. Then, the extracted features are flattened and concatenated to have them as a sequence. A learnable order embedding is added to this sequence. Next, the sequence is fed to the first transformer encoder, which will feed to the next one the processed sequence (there are T transformer encoders). The architecture of a transformer encoder is presented on the right of Figure 6.3; it consists of a multiheaded self-attention (MSA) and a feed-forward network (FFN). The FFN has various FC layers with ReLU activation. The last transformer encoder outputs a sequence with the same shape and order as the original one that goes through a Softmax layer to have a probabilistic meaning. Thus, the output indicates for each patch the probability that it contains stenosis. For more details, please refer to their paper.
They provide an implementation of their network on *GitHub* that is flexible regarding the number and the size of the patches. Nevertheless, it had to be modified further because our input uses 2D patches. The main impact is on the convolution (3D kernels to 2D kernels) and the dimensions of the Einstein sums they used. In order to fight overfit, a dropout layer has been added at the end of the CNN block. Despite this, our network is very similar to theirs.

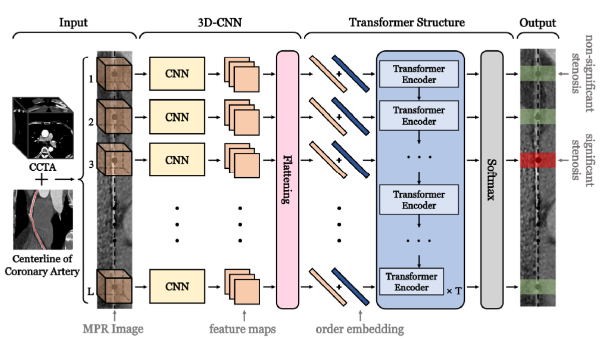


Figure 6.2: Transformer network from [18]. The CNN and transformer layers are presented on the right (figure 6.3). **The figure comes from their paper [18].**

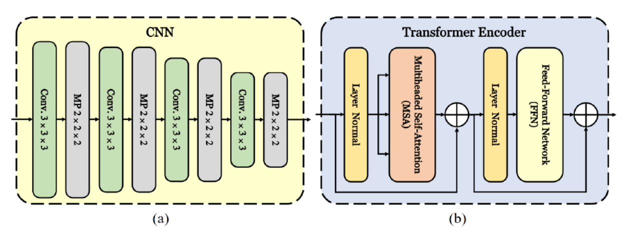


Figure 6.3: Main layers of the TR-Net proposed by [18], which architecture is presented on the left (figure 6.2). **The figure comes from their paper [18].**

6.2.2 Transformer without patient data

The TR-Net was first implemented inside the general architecture (figure 3.2) without the patient information. The resulting architecture is illustrated in figure 6.4. The sequence of patches of each view goes to a TR-Net. Then, the TR-Net outputs the MI prediction for each patch; thus, no further common analysis can be made. So, the prediction at the artery level is simply the maximum prediction of all the patches and the prediction at the patient level is the maximum prediction of all the arteries.
As mentioned previously, the dataset contains patients with missing arteries. When facing them, their missing artery patches are replaced by void patches, and the network is used normally. The losses coming from the missing arteries are not added to the total loss, and the blocks of the missing arteries are frozen during backpropagation.

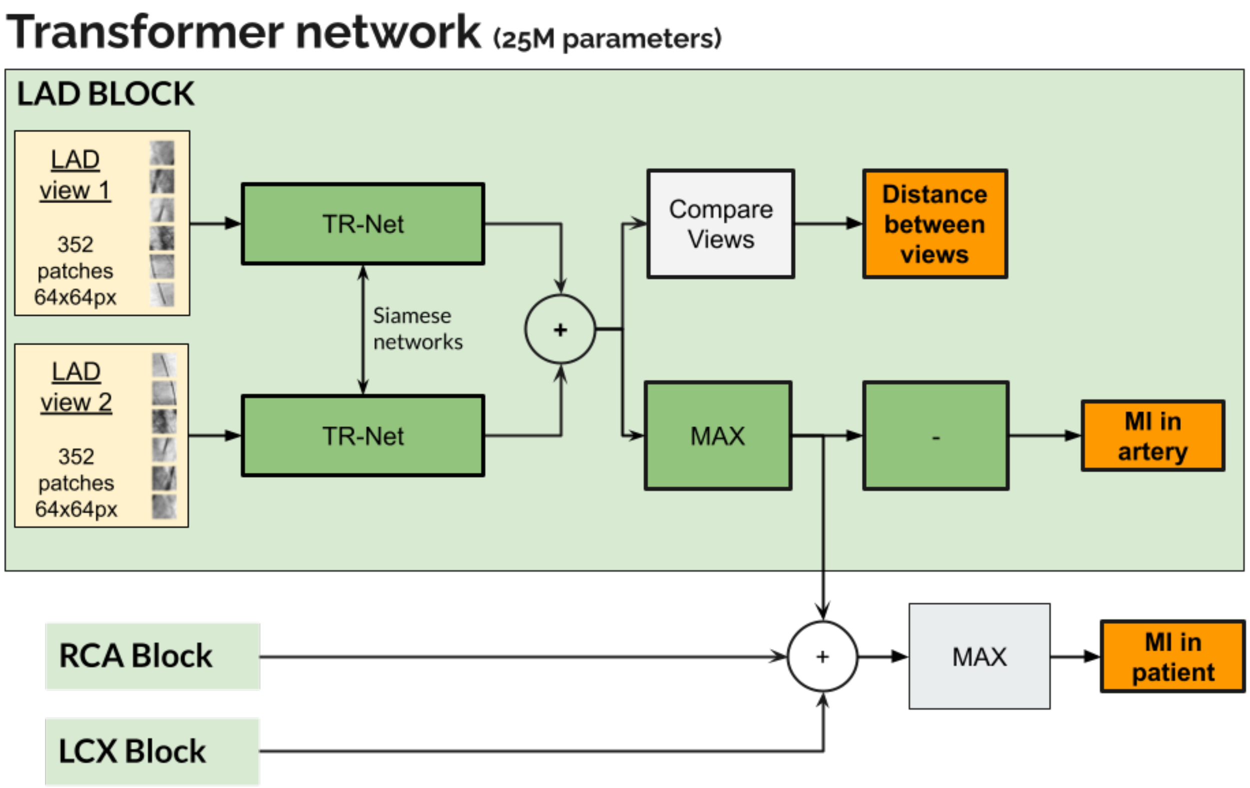


Figure 6.4: Architecture of the Transformer network. The network aims to predict future MI for a patient, given patches extracted along arteries of his/her CA images. The sequences are processed by Transformers. The "plus" blocks symbolise concatenation operations. The "-" inside of a block indicates that it does nothing to the data.

6.2.3 Transformer with patient data added in the sequence

Next, we added the patient data to see how it could improve the predictive performance. To do so, the patient data analysis ANN network implemented in section 4.1.2 is used in the "Information Block". The patient features that will be used in the transformer are the 10 features extracted by the ANN network after its second FC layer. The patient features are added to the flattened sequence of features extracted from the patient patches. Then, the transformer network is used as presented previously. The full architecture is presented in figure 6.5, TR-Net is only drawn once, but both blocks are identical.

This way of using patient data and images in transformers is easy to implement; however, it alters the idea of a sequence of images going from the start to the end of the artery. This implementation is motivated by [48], they concatenate their metadata (age and gender of the patient) to the features extracted from ECG through CNN and LSTM. Then, they feed the whole sequence to their transformer. This network is then used to predict arrhythmia.

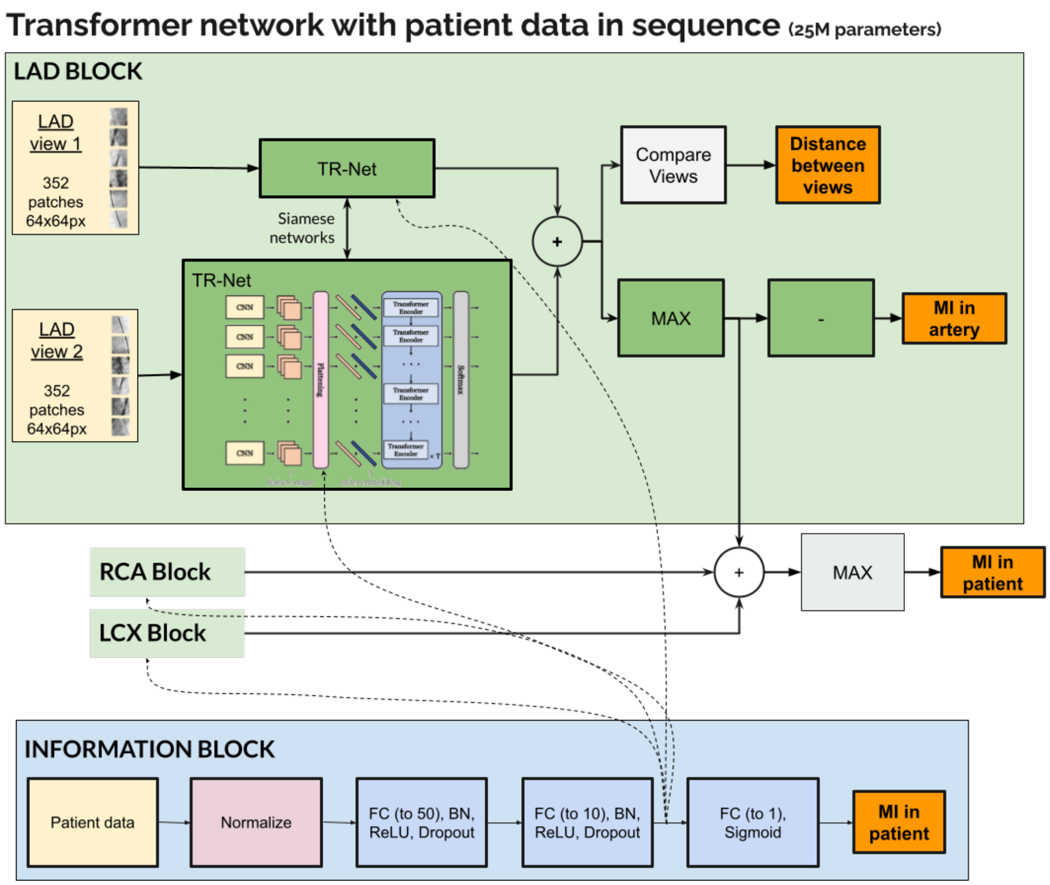


Figure 6.5: Architecture of the Transformer network with patient data added to the sequence of patches. The network aims to predict future MI for a patient, given his/her basic information and patches extracted along the arteries of his/her CA images. The sequences are processed by Transformers, and the patient features are added to the sequence of features. The "plus" blocks symbolise concatenation operations. The "-" inside of a block indicates that it does nothing to the data. TR-Net is only drawn one time, but both blocks are identical. The features from the ANN are concatenated to the sequence of features extracted from patches.

6.2.4 Transformer with patient data added in the softmax

As previously mentioned, adding the patient data to the sequence is unnatural as the sequence should represent the evolution of the artery's patches features. Thus, another approach has been considered: adding the patient data just before the classification softmax layer (which contains some FC layers). The architecture is shown in figure 6.6. This approach is closer to what has been proposed for CNN, where features from two sources are concatenated for further analysis. All other parts are implemented as previously. Again, TR-Net is only drawn once, but both blocks are identical.

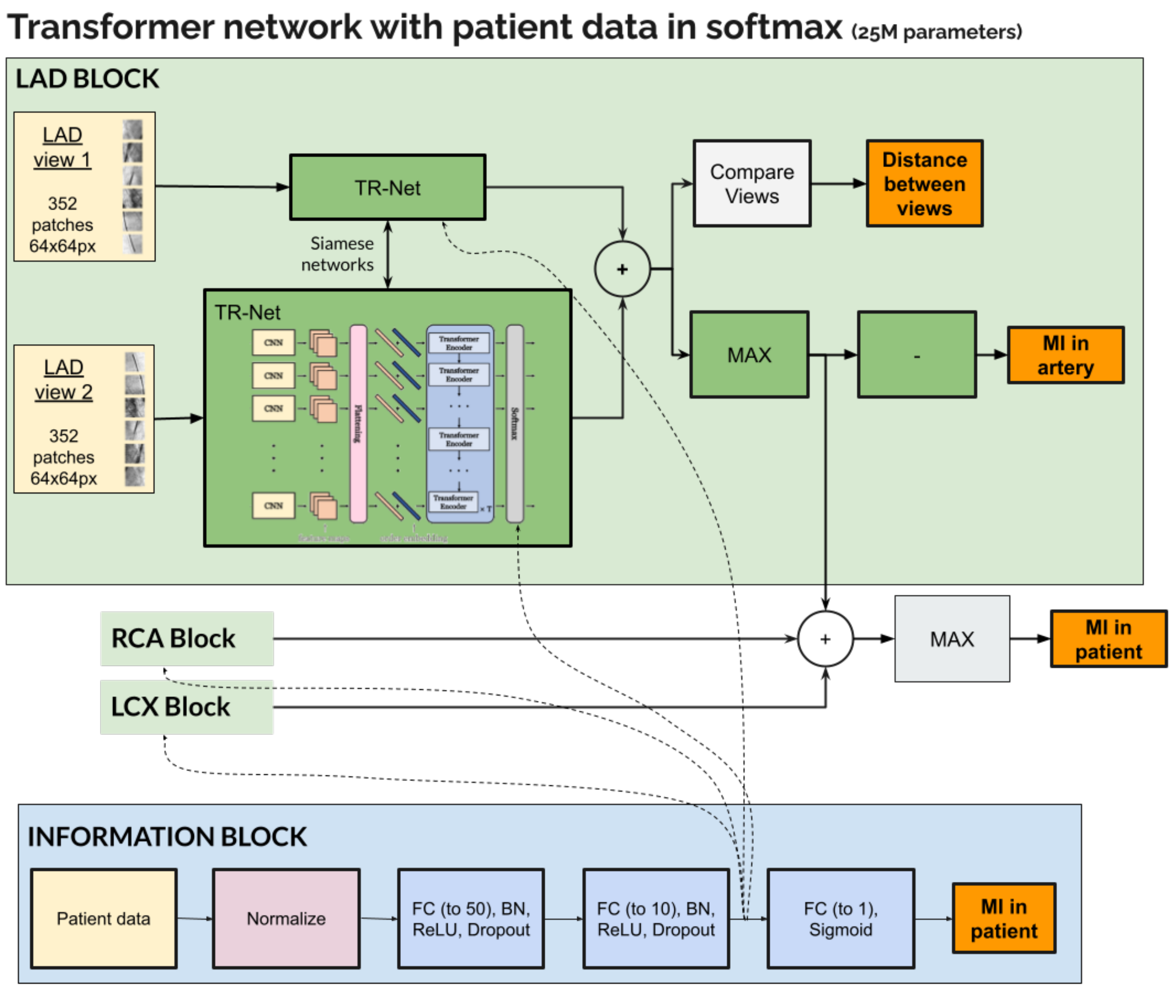


Figure 6.6: Architecture of the Transformer network with patient data added to the softmax layer. The network aims to predict future MI for a patient, given his/her basic information and patches extracted along the arteries of his/her CA images. The sequences are processed by Transformers, and the patient features are added at the beginning of the Softmax layer. The "plus" blocks symbolise concatenation operations. The "-" inside of a block indicates that it does nothing to the data. TR-Net is only drawn one time, but both blocks are identical. The features from the ANN are concatenated to the features before the FC layers that lead to the softmax activation.

6.3 Results

6.3.1 Configuration

The networks have always been trained during 30 epochs with a batch size of 4. A scheduler divides the learning rate by 10 if no improvement is detected after 5 epochs. The weights are initialised with a Xavier uniform distribution [44]. The training duration is shorter than CNN but still significant; a 5-fold validation may last between six and eight hours. The list of the HP present during the grid search is provided in appendix A.1.3. The transformer without patient data has been once evaluated with the "Complete patients", i.e. only the patient with the three arteries available. The transformer without patient data has been trained with AUC but with some pretraining steps done with BCE ("BCE pretraining steps"). In such a situation, during the 10 first epochs, the network was trained on BCE loss with SGD before switching to AUC with PESG. To have a fair comparison, it lasted 40 epochs and not 30. Another approach was to load at the start of the training a network which had been trained with BCE and then train it with AUC, referred to as "BCE trained preloaded". For the transformer network with patient data in softmax, it has been considered with two ranges of the artery loss ratio, one that lets the grid search free (vanilla) and another that enforces high weight for the arteries' losses ("Big ratio for artery losses").

6.3.2 Performance on the clinical data

Table 6.1 shows the best performance obtained by the different models. The F1-Score at the artery level and at patient level are provided. As for CNN, the score at the artery level has to be considered with care as there is only a small number of MI for each artery. The transformer architecture (section 6.2.2) is referred as "Transformer", the second one, with patient data, as "Transformer - Sequence" (section 6.2.3) and the last one as "Transformer - Softmax" (section 6.2.4). All have been evaluated in a different configuration (loss, sampling method, ...). The evolution of their F1-Score along epochs is provided in appendix A.4.2.

Table 6.1: Performance of the transformer architectures to predict MI from CA images (and patient information). The "Transformer" is presented in 6.2.2, the "Transformer: Sequence" in 6.2.3 and the "Transformer: Softmax" in 6.2.4.

Transformer Model	Balance	Pred Loss	Other	F1 (valid) (mean ± std)	Mean artery F1 valid (LAD / LCX / RCA)	AUC-ROC (valid) (mean ± std)
Always no-MI	-	-	-	0.00	0.00 / 0.00 / 0.00	0.50
Random probabilistic	-	-	-	0.0858	0.06 / 0.04 / 0.03	0.50
Transformer	over	BCE	Only with complete patients	0.19 ± 0.09	0.11 / 0.03 / 0.00	0.59 ± 0.11
	over	BCE	-	0.25 ± 0.14	**0.14** / **0.13** / 0.00	0.61 ± 0.09
	over	Focal	-	0.22 ± 0.05	0.07 / 0.00 / 0.04	0.62 ± 0.06
	over	AUC	BCE pretraining steps	0.20 ± 0.04	0.09 / 0.10 / **0.07**	0.62 ± 0.08
	over	AUC	BCE trained preloaded	0.22 ± 0.16	0.02 / 0.02 / 0.02	0.61 ± 0.08
Transformer: Sequence	over	BCE	-	0.20 ± 0.06	0.11 / 0.07 / 0.00	0.58 ± 0.06
Transformer: Softmax	over	BCE	-	0.22 ± 0.08	0.08 / 0.03 / 0.00	0.61 ± 0.10
	over	BCE	Big ratio for artery losses	**0.27 ± 0.04**	0.12 / 0.10 / 0.00	**0.67 ± 0.05**

6.4 Discussion

6.4.1 Transformers on patches

Using all the patients with at least one artery available, and not just those with all the arteries available, improves the performance significantly. It also improves the performance at the artery level. The focal loss seems to be less well suited than the BCE loss as its utilisation decreases the performance in both patient and artery prediction.
Training this network with AUC was not easy as it could easily have an exploding loss. That is the consequence of the transformers being a bit "hard" to train and the AUC loss difficulty in early epochs. Thus, the HP had to be selected with great care. Specifically, it was impossible to train the network without previous epochs on a BCE loss. Pretraining the network results in lower performance than loading a pretrained network, but with a much lower standard deviation. That may be a consequence of some residuals learnings of the preloaded network. More, the preloaded networks perform poorly at the artery level. Thus, pretraining seems to be a better approach. Due to the sensibility of the AUC training and because the best performance is achieved with BCE, it was decided to continue with BCE loss for the transformer with patient data.

6.4.2 Transformers on patches and patient data

Adding the patient features in the softmax seems more efficient than adding them inside the sequence, as the sequence method reaches the lowest performance for the global MI detection (but a similar performance for the artery level performance). It makes sense as adding the patient features to the sequence is not very meaningful, as already discussed. Enforcing high weights on the artery prediction seems to have a good influence on the performance; the global F1-Score gains 0.05 and loses standard deviation. The prediction at the artery level also logically increases.

6.4.3 Architectures comparison

All the methods achieve a better prediction at the patient level than the naive one. Most of them equalise or beat the naive strategies at the artery prediction level. The transformer obtains the best performance with patient data in softmax. That shows that the network can use the images and the patient data to improve its performance. One could have expected more improvement, mainly because the reached performance is smaller than the performance from patient data only (ANN only reaches 0.30, see table 4.2). That implies that more exploration should be done to find the correct way to add the patient information and that the network does not use all the patient features' explanation potential.
Regarding the artery level prediction performance, the different networks tend to use a bit of the three arteries and not focus only on one. The higher performance is on LAD, then LCX and finally RCA, which makes sense as there are more examples of LAD than LCX, and more of LCX than RCA. For the artery level, the best performing network is the transformer trained on BCE without patient features.
Despite this analysis, most of the results achieve deceptive performances. There are at least four explanations. First, the input provided is relatively poor (as explained above). Second, TR-Net is implemented for a more manageable task (stenosis) while using richer data (3D patches, correct centerline, ...). Thus this network may not have enough capacity for the challenge provided. Third, the exact location of the MI is not provided, and the network has to learn from an approximate location, despite providing a prediction at the patch level. Fourth, the patients who miss arteries complicate the learning process.

6.4.4 Interpretability

For the Transformers, only the Saliency map has been considered; the result is presented in figure 6.7. It has been applied on the best performing network: Transformer with patient data added in softmax 6.6. For each artery, the Saliency maps of the ten patches with higher activation have been inspected, and the three more interesting ones are presented. This figure has two outcomes. First, the network mainly focuses on the arteries and not on the background or artefacts (but that doe not mean that it will predict MI correctly). Second, even when the patch extraction is poor (i.e. when the artery is only partially on the image), the network detects it. However, without the entire width of the artery, MI prediction may be complicated as that's the shrinkage of the artery that may cause it.

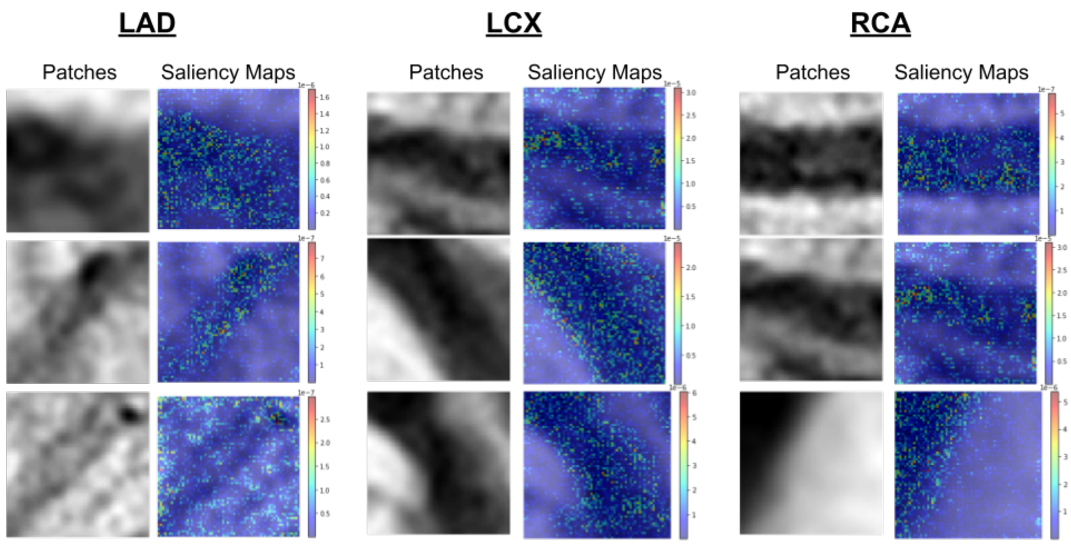


Figure 6.7: Interpretability analysis of the Transformer with patient data in softmax architecture 6.6. Three out of the ten patch with the higher attention have been selected for each artery. Their Saliency map (introduced in section 3.3.5) is displayed on their right.

7. Discussion

Table 7.1 reminds the main results obtained during the k-fold evaluations. Table 7.2 indicates the performance on the testing dataset for: an interventional cardiologist, the Random Probabilistic model, the best CNN (section 5.2.3) and the best Transformer (section 6.2.4). Note that these results are just *indicative* (see 7.1.5).

Table 7.1: Summary of the performances of the most relevant models presented in this work. The performance is given by F1-Score (at patient and arteries level), AUC-ROC, precision and recall (at patient level).

Architecture	Model	F1 (valid) (mean±std)	Mean artery F1 (valid) (LAD/LCX/RCA)	AUC-ROC (valid) (mean±std)	Precision (valid) (mean±std)	Recall (valid) (mean±std)
Naive	**Random Probabilistic**	~0.1	0.06 / 0.04 / **0.03**	0.50	0.1	0.1
Patient Data	**ANN (BCE to AUC)**	0.30 ± 0.09	-	**0.67** ± 0.04	0.20 ± 0.08	**0.70** ± **0.11**
CNN	**Common (BCE)**	0.31 ± 0.07	0.00 / 0.00 / 0.00	0.65 ± 0.03	0.25 ± 0.08	0.50 ± 0.15
	Max (AUC)	0.30 ± 0.11	0.00 / 0.14 / 0.00	0.64 ± 0.04	0.21 ± 0.13	0.69 ± 0.11
	Max + patient data (AUC - normal)	**0.36** ± **0.12**	0.11 / **0.28** / 0.00	**0.67** ± 0.04	**0.36** ± **0.18**	0.44 ± 0.10
Transformer	**Without patient data (BCE)**	0.25 ± 0.14	**0.14** / 0.13 / 0.00	0.61 ± 0.09	0.35 ± 0.34	0.30 ± 0.20
	Patient data in Softmax (BCE - Big artery ratio)	0.27 ± 0.04	0.12 / 0.10 / 0.00	**0.67** ± 0.05	0.17 ± 0.03	0.63 ± 0.13

Table 7.2: Performance obtained on the testing dataset for: the Random Probabilistic, an interventional cardiologist, the best CNN (6.2.4) and the best Transformer (6.2.4). **These results are *indicative* (section 7.1.5).**

Predictor	F1-Score (test)	AUC-ROC (test)	Precision (test)	Recall (test)	Specificity (test)
Random Probabilistic	0.058	0.500	0.058	0.058	**0.942**
Interventional Cardiologist	0.095	0.484	0.054	0.400	0.568
CNN	0.167	0.627	0.10	0.600	0.654
Transformer	**0.412**	**0.722**	**0.304**	**0.636**	0.807

7.1 Our approaches

7.1.1 Patient level

From table 7.1, one can see that all our methods outperform the naive approach at the patient level. The transformers network shows a lower F1-Score than CNN and ANN. The Transformer using the patient data reaches a lower score than the patient data alone. Five facts may explain this low performance: (i) the imbalance is higher in the transformer dataset than in the others, (ii) the input given to the network is of mediocre quality (as explained in its chapter); (iii) the network is trained with partially complete patients; (iv) the main block (the TR-Net) has been developed for a more manageable task and with richer data, and (v) the MI is only provided at the section level, despite the Transformer predicting MI at the patch level.
The patient data alone reaches a performance comparable to the performances of the CNN with CA alone. That is surprising as one may expect that the images contain more information than patient data. Nevertheless, in the current situation, the different dataset's challenges may complicate the network's learning from images. Adding the patient information to the CNN beats both the CNN and the patient data, indicating that more information can be extracted from both. Finally, at the patient level, the CNN with patient data shows the best performance with respect to the F1-Score.

7.1.2 Artery level

At the artery level, the situation is different. None of our methods can beat the random classifier for the RCA artery. That may be explained by the small number of positive MI cases in this artery (only 12 before splitting between test-train-validation and the k-folds). Interestingly, most of the networks show better performance on LCX than LAD. However, there are more examples of MI in LAD than in LCX. That may indicate that MI on LCX is more accessible to detect than on LAD. The only model that performs better on LAD than LCX is the with high weight on the artery loss. In this situation, the network has to learn the most abundant MI class to reach a lower loss. The Transformer can use both LAD and LCX compared to most CNN.

7.1.3 Other metrics

The findings for the AUC metric are similar to those of F1-Score: both metrics are related. Except for the Transformer with patient data that achieves a similar AUC performance to CNN with patient data despite a significantly lower F1-Score. In medical applications, recall is more important than precision, as missing a detection could be fatal. The recall is good for the ANN, the CNN with Max and the Transformer with patient data. The network with the best F1-Score (CNN with patient data) reaches a low recall score, which may be an issue in a real-world application. Regarding these metrics, the Transformer with patient data does not reach as bad performances as one could assume from the F1-Score only.

7.1.4 Interpretability

The CNN interpretability analysis pointed out that the network mainly looks at the artery, not the background. Sometimes, it looked like the network was giving more attention to MI sections. However, these findings are questionable, and there is a significant disparity between the arteries. The Transformer also looks mainly at the artery for the most activated patches. A human would say that the Transformer understood better what an artery is. However, it is much easier for it do detect them than for the CNN: the CNN looks at a big image with lots of artefacts and background, whereas the Transformer only has a small patch whose main element is often the artery itself.

7.1.5 Performance on testing dataset

Table 7.2 shows the performance on the testing dataset. These results have to be considered with great care for several reasons. First, the testing dataset contains very few positive cases (just by correctly classifying one more MI as positive than negative, the random classifier would have reached an F1-Score of 0.23 instead of 0.06). Second, the dataset used is not the same for the Transformer (it has fewer available patients due to its specific architecture). Third, the doctor had less information at his disposal. He only had access to the raw images, whereas the networks also used the patient information and additional insights (masks, centerline, ...). More, the performance is only provided for one doctor. For these reasons, this table has to be considered as an *indication* more than a result. That is further motivated by the fact that most of the results obtained are out of the confidence interval computed from the validation dataset.
The CNN and the Transformer outperform the naive model for all the metrics, except for the specificity. Interestingly, the Transformer achieves better performance than the CNN on this dataset and much higher performance than in validation. Of course, the interventional cardiologist's performance outperforms the random classifier; in particular, his recall is much higher. The performance he achieves remind us how difficult is the current task, even for a trained physician. The CNN and the Transformer achieve a better performance than him on this dataset (but also when looking at the performance on validation, table 7.1). Thus, Machine Learning (and in particular, Deep Learning) can definitively help regarding future MI prediction. Due to the uncertainty of the testing dataset, we may not assert that the Deep Learning architectures outperform the physician, but we can say that they, at least, reach comparable performances.

7.2 Previous works

As mentioned previously (section 1.2), no literature aims in predicting *future* MI from XCA images with Deep Learning. Nor does a record of physicians' performances exist for this specific task (we only have the previously presented doctor's performance). Thus, it is not possible to judge the quality of the results. However, comparing with other similar works will show that our results are coherent with what has been done.

7.2.1 Stenosis detection

The results obtained by other works for the stenosis prediction are much better than ours, [10] obtained an F1-Score of 0.91, [12] reached an AUC-ROC of 0.862 and the TR-Net [18] that is used as the backbone of our transformer network reaches an F1-Score of 0.79. However, these performances can not be directly compared to our network's performances because the future MI prediction is a more complex task than stenosis detection. As a reminder, the last Master Thesis [4] had relatively poor results for the MI task (F1-Score of 0.22) but achieved decent performance for stenosis (F1-Score of 0.66) with the *same* network. Thus, the performance of stenosis networks can not be compared to ours.

7.2.2 Culprit lesion

In [21], future MI was predicted from patches centred on lesions from XCA images. This task is closer to ours. The performance of their network is an F1-Score of 0.571 and a recall of 0.667. Their task is a bit easier than ours for two reasons. First, the localisation was much easier as the patch was small and centred on the lesion. Second, they only had two classes, "lesion" and "culprit lesion". Nevertheless, in the current work case, the network has to separate "culprit lesion" from "lesion" and "no lesion". In their work, they provide a cardiologist performance (F1-Score 0.348 & recall 0.444). Our best model outperforms the specialist in F1-Score and achieves a similar recall score. That is motivating as it may again indicate that our networks *may* perform similarly to a physician.

7.2.3 MI from patient data

MI prediction from patient data reaches very high performance for *close* MI. However, as already discussed, this can not be compared to *future* MI. Papers that consider predicting future MI from patient data reach somewhat similar performances to our networks, [8] reaches an AUC of 0.83 but an F-Score of 0.101 and [9] reaches an AUC of 0.72. These two works are comparable to what is achieved by our ANN on patient data and our other networks. That again points out the images' low predictive performance compared to the patient data.

7.2.4 MI from XCA

Finally, the previous Master Thesis [4] tackled the same challenge as us and with the same data. It can be compared more safely to our performance. The main difference between the two works is that the previous one considered the artery section level, whereas the whole patient level has been considered here. That has two main consequences. First, each prediction has access to more data in this work than in the previous one. Rather than having only one view of one artery, the input has the two views of the three arteries and, sometimes, patient data. Thus, it is assumed (and it was the objective of this work) that this additional information will increase the predictive performances. Second, the drawback of this richer input is a smaller dataset than previously available. So, despite the additional information, training the network may be more demanding. The previous Master Thesis: obtained F1-Score 0.22, precision 0.25 & recall 0.22. The difference between our performances and his demonstrates that the additional information improved the predictive performance, thus, that the patient level is beneficial.

7.2.5 MI from CCTA

Future MI prediction from CCTA images, as done in [19], reaches a near AUC than our methods (0.70) while having a similar recall (0.659) to our ANN, CNN-Max and Transformer with patient data. However, future MI prediction is not their primary objective, and they do not use the same data type.

8. Conclusion

8.1 Summary

The main objective of this work was to detect future Myocardial Infarction (MI) from a dataset of patient data and X-ray Coronary Angiography images (CA). Both were used separately and conjointly. To tackle this challenge, different methods have been used:

- Prediction of MI with traditional Machine Learning algorithms from the patient data;
- Prediction of MI with Convolutional Neural Networks (CNN) from CA, and patient features;
- Prediction of MI with Transformers from patches along the artery from CA, and the patient features.

The main challenges faced were the class imbalance, the limited data, and the heterogeneity of the different artery sections. Different state-of-the-art methods were used to deal with them. The best performance in F1-Score was obtained by the CNN (F1-Score: 0.36±0.12; AUC-ROC: 0.67±0.04; Precision: 0.36±0.18 & Recall: 0.44±0.10). The prediction from the patient data only, through an Artificial Neural Network (ANN), also achieved competitive performance (F1-Score: 0.30±0.09; AUC-ROC: 0.67±0.04; Precision: 0.20±0.08 & Recall: 0.70±0.11). The Transformer reached a lower F1-Score but is still competitive regarding AUC-Score and recall, which is relevant for medical applications (F1-Score: 0.27±0.04; AUC-ROC: 0.67±0.05; Precision: 0.17±0.03 & Recall: 0.63±0.13). Moreover, for the Transformer, there is still room for improvement and their interpretability analysis shows promising behaviours.
It is hard to assess these results' quality as no paper worked precisely on the same challenge. However, the results are coherent with similar works, and the interpretability analysis shows some exciting behaviours from the networks. More, the networks reach similar performance as interventional cardiologist. Despite that, more work is needed before considering implementing these networks in real-world scenarios. In the next section, many improvements and new ideas are proposed to face this exciting challenge.

8.2 Future works

8.2.1 Dataset improvement

For all the approaches, a more extensive dataset with more cases of Myocardial infarction (MI) should help. For the Transformers, a more specific dataset could be provided. The patches extraction strategy presented in this work suffers the lack of a centerline of good quality. More, having a more precise location of the MI should definitively help the network (currently, it only knows if, among all the patches of a given section, one has MI, despite providing a prediction at the patch level). With both a proper centerline and a local MI information, the box size heterogeneity would not be an issue anymore as one could extract the patches along the centerline of the main image.

8.2.2 Metadata and images features aggregation

The extracted features of CA and patient data have been concatenated and processed further. This approach was motivated by different previous works ([41], [42], [39] & [43]). However, other papers propose alternatives. For example, one paper proposes to use an attention mechanism so that the patient data applies attention to the features extracted by the CNN [49]. This approach sounds very promising as features from both sources are not just merged but have a direct influence on each other. Our aggregation strategies are questionable for the Transformers, and alternatives may be proposed.

8.2.3 Graph Convolutional Networks

A step ahead of Transformer would be to use Graph Convolutional Networks (GCN). Such a network could use the same idea of a list of patches as the Transformers, which decreases the amount of background in the input. Nevertheless, in addition, it could enhance the connectivity between the patches.
Some of the weaknesses of the current Transformer implementation are that it does not consider bifurcations and the distance between patches. Currently, bifurcations are considered in series and not in parallel; GCN could tackle that by creating a new branch. Our Transformer has to learn the concept of proximity between patches, whereas a GCN would already embed this information. GCN is thus promising, but requires an improvement in the input extraction, similarly to the one discussed above for the Transformers.
Applications of GCN to Coronary Arteries exist. In [50], the authors propose to use this technology to segment the coronary computer tomography angiography (CCTA) and improve the stenosis detection (which is related to MI). Coronary plaques are localised on CCTA thanks to GCN in [51]: they use a transformer and provide it with a particular graph-like patch embedding. More formal work on the relationship between Transformers and GCN can be found in [52], where they provide a generalisation of the Transformers to graphs.

8.2.4 Anomaly Detection Network and Multiple Instance Learning

Anomaly Detection networks are, briefly, networks that learn the distribution of a set and are then used to detect samples that are far from this distribution (anomalies). This approach is attractive as it only requires samples of the most frequent class. However, they may show poor performance in samples relatively close to the normal distribution. Such networks could be considered an alternative for our challenge, facing the limited number of positive samples.
In [53], the authors use an Anomaly Detection Network to detect polyps from colonoscopy; they propose an architecture that also uses some anomalies during training (Few-shot Anomaly Detection), which tackles the main challenge of this kind of architecture. In [54], they use abnormality detection to find stenosis from CCTA images. From a U-Net-like architecture, they grade the reconstruction of the image to infer potential stenosis.
The Multiple Instance Learning approach involves working with a bag of samples and, instead of classifying sample per sample, classifying the bag. The bag is considered positive if any of its elements is positive. A good introduction and implementation for Multiple Instance Attention-based Neural Networks is provided in [55]. The bag, in our case, could be the set of patches of a section and the label if the artery section contains or not MI. With this technology, in opposition to the Transformer or GCN approach, the absence of precise MI labelling would no longer be a challenge.
Such architecture has been used to predict significant stenosis from CCTA images in [56]. With great success, MIL with Neural Networks has also been used to detect MI from ECG devices [57] [58]. The performance is excellent, but they do not indicate how much they predict the MI in advance. Despite not being medical, a paper [59] proposes an exciting network that merges anomaly detection and MIL (available on *GitHub*).

Bibliography

[1] ABUBAKAR, I. I., TILLMANN, Taavi, et BANERJEE, Amitava. Global, regional, and national age-sex specific all-cause and cause-specific mortality for 240 causes of death, 1990-2013: a systematic analysis for the Global Burden of Disease Study 2013. Lancet, 2015, vol. 385, no 9963, p. 117-171.

[2] SHIN, Hoo-Chang, ROTH, Holger R., GAO, Mingchen, et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE transactions on medical imaging, 2016, vol. 35, no 5, p. 1285-1298.

[3] AUER, Reto, GENCER, Baris, RÄBER, Lorenz, et al. Quality of care after acute coronary syndromes in a prospective cohort with reasons for non-prescription of recommended medications. PloS one, 2014, vol. 9, no 3, p. e93147.

[4] VUILLECARD, Pierre. Myocardial infarction prediction from angiographies based on deep learning, 2022.

[5] NAG, Procheta, MONDAL, Saikat, AHMED, Foysal, et al. A simple acute myocardial infarction (Heart Attack) prediction system using clinical data and data mining techniques. In : 2017 20th International Conference of Computer and Information Technology (ICCIT). IEEE, 2017. p. 1-6.

[6] KOJURI, Javad, BOOSTANI, Reza, DEHGHANI, Pooyan, et al. Prediction of acute myocardial infarction with artificial neural networks in patients with nondiagnostic electrocardiogram. Journal of Cardiovascular Disease Research, 2015, vol. 6, no 2.

[7] THAN, Martin P., PICKERING, John W., SANDOVAL, Yader, et al. Machine learning to predict the likelihood of acute myocardial infarction. Circulation, 2019, vol. 140, no 11, p. 899-909.

[8] MANDAIR, Divneet, TIWARI, Premanand, SIMON, Steven, et al. Prediction of incident myocardial infarction using machine learning applied to harmonized electronic health record data. BMC medical informatics and decision making, 2020, vol. 20, no 1, p. 1-10.

[9] GUPTA, Shagun, KO, Dennis T., AZIZI, Paymon, et al. Evaluation of machine learning algorithms for predicting readmission after acute myocardial infarction using routinely collected clinical data. Canadian Journal of Cardiology, 2020, vol. 36, no 6, p. 878-885.4

[10] OVALLE-MAGALLANES, Emmanuel, AVINA-CERVANTES, Juan Gabriel, CRUZ-ACEVES, Ivan, et al. Transfer learning for stenosis detection in X-ray coronary angiography. Mathematics, 2020, vol. 8, no 9, p. 1510.

[11] HE, Kaiming, ZHANG, Xiangyu, REN, Shaoqing, et al. Deep residual learning for image recognition. In : Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 770-778.

[12] AVRAM, Robert, OLGIN, Jeffrey E., WAN, Alvin, et al. CathAI: fully automated interpretation of coronary angiograms using neural networks. arXiv preprint arXiv:2106.07708, 2021.

[13] CHOLLET, François. Xception: Deep learning with depthwise separable convolutions. In : Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 1251-1258.

[14] LIN, Tsung-Yi, GOYAL, Priya, GIRSHICK, Ross, et al. Focal loss for dense object detection. In : Proceedings of the IEEE international conference on computer vision. 2017. p. 2980-2988.

[15] RONNEBERGER, Olaf, FISCHER, Philipp, et BROX, Thomas. U-net: Convolutional networks for biomedical image segmentation. In : International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015. p. 234-241.

[16] YANG, Su, KWEON, Jihoon, ROH, Jae-Hyung, et al. Deep learning segmentation of major vessels in X-ray coronary angiography. Scientific reports, 2019, vol. 9, no 1, p. 1-11.

[17] XIE, L., ZHANG, H., LIU, X., et al. Automatic and multimodal analysis for coronary angiography: training and validation of a deep learning architecture. Eurointervention: Journal of Europcr in Collaboration with the Working Group on Interventional Cardiology of the European Society of Cardiology, 2020.

[18] MA, Xinghua, LUO, Gongning, WANG, Wei, et al. Transformer Network for Significant Stenosis Detection in CCTA of Coronary Arteries. In : International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Cham, 2021. p. 516-525.

[19] LIN, Andrew, MANRAL, Nipun, MCELHINNEY, Priscilla, et al. Deep learning-enabled coronary CT angiography for plaque and stenosis quantification and cardiac risk prediction: an international multicentre study. The Lancet Digital Health, 2022, vol. 4, no 4, p. e256-e265.

[20] SHI, Xingjian, CHEN, Zhourong, WANG, Hao, et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems, 2015, vol. 28.

[21] THANOU, Dorina, SENOUF, Ortal Yona, RAITA, Omar, et al. Predicting future myocardial infarction from angiographies with deep learning. 2021.

[22] GIRSHICK, Ross, DONAHUE, Jeff, DARRELL, Trevor, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In : Proceedings of the IEEE conference on computer vision and pattern recognition. 2014. p. 580-587.

[23] LECLERC, Guillaume, ILYAS, Andrew, ENGSTROM, Logan, et al. FFCV library (`https://github.com/libffcv/ffcv/`)

[24] VAN RIJSBERGEN, Cornelius Joost. Information retrieval. 2nd. newton, ma. 1979.

[25] YUAN, Zhuoning, YAN, Yan, SONKA, Milan, et al. Large-scale Robust Deep AUC Maximization: A New Surrogate Loss and Empirical Studies on Medical Image Classification. arXiv preprint arXiv:2012.03173, 2020.

[26] GUO, Zhishuai, YUAN, Zhuoning, YAN, Yan, et al. Fast Objective & Duality Gap Convergence for Nonconvex-Strongly-Concave Min-Max Problems. arXiv preprint arXiv:2006.06889, 2020.

[27] SULAM, Jeremias, BEN-ARI, Rami, et KISILEV, Pavel. Maximizing AUC with Deep Learning for Classification of Imbalanced Mammogram Datasets. In : VCBM. 2017. p. 131-135.

[28] SIMONYAN, Karen, VEDALDI, Andrea, et ZISSERMAN, Andrew. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034, 2013.

[29] SUNDARARAJAN, Mukund, TALY, Ankur, et YAN, Qiqi. Axiomatic attribution for deep networks. In : International conference on machine learning. PMLR, 2017. p. 3319-3328.

[30] SELVARAJU, Ramprasaath R., COGSWELL, Michael, DAS, Abhishek, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In : Proceedings of the IEEE international conference on computer vision. 2017. p. 618-626.

[31] ZEILER, Matthew D. et FERGUS, Rob. Visualizing and understanding convolutional networks. In : European conference on computer vision. Springer, Cham, 2014. p. 818-833.

[32] GOODFELLOW, Ian J., SHLENS, Jonathon, et SZEGEDY, Christian. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

[33] LOH, Wei-Yin. Classification and regression trees. Wiley interdisciplinary reviews: data mining and knowledge discovery, 2011, vol. 1, no 1, p. 14-23.

[34] BREIMAN, Leo. Random forests. Machine learning, 2001, vol. 45, no 1, p. 5-32.

[35] CORTES, Corinna et VAPNIK, Vladimir. Support-vector networks. Machine learning, 1995, vol. 20, no 3, p. 273-297.

[36] TOLLES, Juliana et MEURER, William J. Logistic regression: relating patient characteristics to outcomes. Jama, 2016, vol. 316, no 5, p. 533-534.

[37] CHAN, Tony F., GOLUB, Gene H., et LEVEQUE, Randall J. Updating formulae and a pairwise algorithm for computing sample variances. In : COMPSTAT 1982 5th Symposium held at Toulouse 1982. Physica, Heidelberg, 1982. p. 30-41.

[38] FRIEDMAN, Jerome H. Greedy function approximation: a gradient boosting machine. Annals of statistics, 2001, p. 1189-1232.

[39] GESSERT, Nils, NIELSEN, Maximilian, SHAIKH, Mohsin, et al. Skin lesion classification using ensembles of multi-resolution EfficientNets with meta data. MethodsX, 2020, vol. 7, p. 100864.

[40] HE, Kaiming, ZHANG, Xiangyu, REN, Shaoqing, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In : Proceedings of the IEEE international conference on computer vision. 2015. p. 1026-1034.

[41] ELLEN, Jeffrey S., GRAFF, Casey A., et OHMAN, Mark D. Improving plankton image classification using context metadata. Limnology and Oceanography: Methods, 2019, vol. 17, no 8, p. 439-461.

[42] THOMAS, Spencer A. Combining Image Features and Patient Metadata to Enhance Transfer Learning. In : 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 2021. p. 2660-2663.

[43] APOSTOLOPOULOS, Ioannis D., APOSTOLOPOULOS, Dimitris I., SPYRIDONIDIS, Trifon I., et al. Multi-input deep learning approach for cardiovascular disease diagnosis using myocardial perfusion imaging and clinical data. Physica Medica, 2021, vol. 84, p. 168-177.

[44] GLOROT, Xavier et BENGIO, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In : Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2010. p. 249-256.

[45] ANDRA, Tache Irina. Contour and Centreline Tracking of Vessels from Angiograms using the Classical Image Processing Techniques. arXiv preprint arXiv:1707.03710, 2017.

[46] RUMELHART, David E., HINTON, Geoffrey E., et WILLIAMS, Ronald J. Learning representations by back-propagating errors. nature, 1986, vol. 323, no 6088, p. 533-536.

[47] HOCHREITER, Sepp et SCHMIDHUBER, Jürgen. Long short-term memory. Neural computation, 1997, vol. 9, no 8, p. 1735-1780.

[48] LE, Minh Duc, RATHOUR, Vidhiwar Singh, TRUONG, Quang Sang, et al. Multi-module Recurrent Convolutional Neural Network with Transformer Encoder for ECG Arrhythmia Classification. In : 2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI). IEEE, 2021. p. 1-5.

[49] PACHECO, Andre GC et KROHLING, Renato A. An attention-based mechanism to combine images and metadata in deep learning models applied to skin cancer classification. IEEE journal of biomedical and health informatics, 2021, vol. 25, no 9, p. 3554-3563.

[50] WOLTERINK, Jelmer M., LEINER, Tim, et IŠGUM, Ivana. Graph convolutional networks for coronary artery segmentation in cardiac CT angiography. In : International Workshop on Graph Learning in Medical Imaging. Springer, Cham, 2019. p. 62-69.

[51] VITI, Mario, TALBOT, Hugues, et GOGIN, Nicolas. Transformer Graph Network for Coronary Plaque Localization in CCTA. In : 2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI). IEEE, 2022. p. 1-5.

[52] DWIVEDI, Vijay Prakash et BRESSON, Xavier. A generalization of transformer networks to graphs. arXiv preprint arXiv:2012.09699, 2020.

[53] TIAN, Yu, MAICAS, Gabriel, PU, Leonardo Zorron Cheng Tao, et al. Few-shot anomaly detection for polyp frames from colonoscopy. In : International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, Cham, 2020. p. 274-284.

[54] FREIMAN, Moti, MANJESHWAR, Ravindra, et GOSHEN, Liran. Unsupervised abnormality detection through mixed structure regularization (MSR) in deep sparse autoencoders. Medical physics, 2019, vol. 46, no 5, p. 2223-2231.

[55] ILSE, Maximilian, TOMCZAK, Jakub, et WELLING, Max. Attention-based deep multiple instance learning. In : International conference on machine learning. PMLR, 2018. p. 2127-2136.

[56] ZREIK, Majd, HAMPE, Nils, LEINER, Tim, et al. Combined analysis of coronary arteries and the left ventricular myocardium in cardiac CT angiography for detection of patients with functionally significant stenosis. In : Medical Imaging 2021: Image Processing. SPIE, 2021. p. 394-401.

[57] FENG, Kai, PI, Xitian, LIU, Hongying, et al. Myocardial infarction classification based on convolutional neural network and recurrent neural network. Applied Sciences, 2019, vol. 9, no 9, p. 1879.

[58] WU, J. F., BAO, Y. L., CHAN, Shing-Chow, et al. Myocardial infarction detection and classification—A new multi-scale deep feature learning approach. In : 2016 IEEE International Conference on Digital Signal Processing (DSP). IEEE, 2016. p. 309-313.

[59] PANG, Guansong, DING, Choubo, SHEN, Chunhua, et al. Explainable deep few-shot anomaly detection with deviation networks. arXiv preprint arXiv:2108.00462, 2021.

[60] YUAN, Zhuoning, GUO, Zhishuai, CHAWLA, Nitesh, et al. Compositional Training for End-to-End Deep AUC Maximization. In : International Conference on Learning Representations. 2021.

A. Appendix

A.1 General

A.1.1 Losses

The different classification losses are further introduced in the next sections.

A.1.1.1 BCE Loss

The first natural choice was the Binary Cross-Entropy (BCE) loss (PyTorch implementation available here).

A.1.1.2 Focal Loss

Due to the imbalanced dataset, other losses have been considered. The focal loss [14] seemed very appropriate as it has been developed to face this challenge. Moreover, it has been used for stenosis detection [12]. Briefly, the Focal Loss is an extension of the Cross-Entropy loss where an additional factor is added that reduces the contribution of easy examples. Formally, the loss is defined as:

$$FL(p_t) = -\alpha_t (1 - p_t)^\gamma \log (p_t) \tag{A.1}$$

p_t is a rewriting of the prediction made by the classifier: it will have a value between, 0 if the classifier is completely wrong, and 1, if it is completely correct. $-\log (p_t)$ corresponds to the rewriting of the standard Cross-Entropy loss. The α_t term is the same as in Balanced Cross-Entropy, where a class receives a higher weight than another. The innovative term is $(1 - p_t)^\gamma$, this term will lead a correct classification loss to zero ($p_t = 1$) and thus reduce the influence of correctly classified samples, whereas the wrongly classified one ($p_t = 0$) will have an unchanged loss. γ defines the intensity of this effect. They provide a numeric example: considering $\gamma = 2$, a sample reaching $p_t = 0.9$ would have a loss one hundred times lower than using classic Cross-Entropy.

A.1.1.3 AUC loss

The previous losses are optimising the accuracy. Some papers propose to optimise other metrics, for example, the AUC-ROC metric [25]. This metric is much closer to the F1-Score (our target) than the accuracy.
Using this loss had some success in medical applications, like classifying imbalanced mammogram dataset [27]. The library from [25] is efficient and simple to implement in a *PyTorch* framework. They highlight the main issues of previous AUC losses: bad influence of easy data and sensitivity to noisy data. The inference of their loss is a bit mathematical and will not be presented here, but it allows us to deal with the two issues above. Note that they convert the computation of the AUC (which needs various samples) to a min-max optimisation sample per sample. A PESG optimiser [26] is used with this loss. They also indicate that training with such loss requires pretraining the network with a more "classic" loss. The same authors propose a new implementation of their AUC loss in a recent paper [60], which does not require a "classical" loss pretraining step. However, it has been considered quickly and did not yield better results.

A.1.2 Interpretability implementation

The interpretability methods are further introduced below, but some implementation details are provided first. Instead of having one image as input, our CNN's receive twelve images, and most libraries do not handle this. More, for all the gradient-based methods, the networks that use a Max function kill the gradient and thus do not provide results for the two less relevant arteries. As a result, the analysis is done artery per artery (the artery block is extracted and only its images are provided, then the interpretability is made). For the same reason, the GradCAM returns a heath-map that is a composition of the entire input, not one for each image. For the Transformer, the implementation was even more complex. Thus, only the Saliency map has been used. Because an inspection of the 352 patches is impossible, only the ten patches with the higher activation were considered. GradCam could not be used because there is no final convolutional layer in the Transformer.

- Saliency map [28]: compute the gradient of the output with respect to the input. Thus, for an image, if a pixel variation is relevant, it will have a high value, else a lower one;
- Integrated gradient [29]: starting from a void image, the input image is recreated step by step, and the gradient is summed at each step. Each pixel will receive an attribution: if it is positive, this pixel contributes to the current prediction, and if it is negative, the pixel goes against it;
- GradCAM [30]: analyses the importance of the different features maps of the last convolutional layer in predicting a particular class. Then, it creates a heath-map from their aggregation of the input's size. A high value means that the pixel is important;
- Input occlusion [31]: iteratively occludes an image region and computes its impact on the prediction. A positive value means the image without this region goes even more toward the prediction;
- FGSM [32]: Fast Gradient Sign Method is a method to generate adversarial images (images similar to the input but that provide the opposite output). It can be used to assess the robustness of a method;

A.1.3 Hyperparameters

The different HP that have been explored during grid searches are listed below. The range used varied depending on the networks and can be found on the *W&B* API. In fact, this API is using a bit clever tool than simple grid search, they call it a "sweep", the documentation is available here.

- Initial learning rate;
- Dropout;
- Weight decay;
- Ratio between *MI in artery* and *MI in patient* losses;
- Ratio between *distance between views* and *MI in patient* losses;
- Ratio between *MI from patient data* and *MI in patient* losses (when patient data used);
- The optimisers HP (momentum for SGD, gamma and margin for PESG);
- The Focal loss HP (when used): alpha, gamma and reduction method.

A.2 Traditional ML

A.2.1 ANN: Performance along epochs

Figures A.1 and A.2, show the evolution of the F1-Score along epochs for the best HP of each ANN training strategy (both on training and validation dataset). These plots show a coherent learning behaviour. There are various overfit behaviours (PESG+AUC, SGD+BCE and SGD+BCE to PESG+AUC): the validation F1-Score decreases in the end, whereas the F1-Score in training continues to increase (or stagnates). With such a "simple" dataset, it is hard to prevent overfitting. For the SGD+BCE to PESG+AUC strategy, the transition to AUC loss and its benefits can be seen: at the epoch 200, the training F1-Score suddenly decreases and then reaches around the same level; however, the testing F1-Score suddenly increases and then stabilises to a higher level than previously. That clearly shows how the AUC loss is much more suited to optimise the F1-Score than the BCE loss. Figures A.3 and A.4 show the evolution of the AUC score along epochs (both on training and validation dataset). The analysis of these plots leads to similar conclusions to those of the F1-Score, which makes sense as the two metrics have strong ties, except that the training AUC continues to increase (and even faster) after the transition in the SGD+BCE to PESG+AUC strategy. The AUC loss optimises this metric; thus, that is normal that the training performance does not decrease.

A.2.2 Influence of the different dataset extensions

Looking at table 4.1, for all the traditional ML models, except Logistic Regression and Naive Bayes, the best performance was obtained with normalised data. That makes sense as normalised data tend to be easier to process for ML methods due to its better distribution and ensures a common scale for all the features. However, Logistic Regression is not sensitive to the scale of the feature; thus, the best performance has been achieved on unnormalised data. It is unclear why Naive Bayes works better on unnormalised data, as it is sensitive to the scale. Except for SVM, all methods obtain their best performance on a balanced (over or under) dataset. Such a dataset avoids having a bias toward a class. Both undersampling and oversampling have their drawbacks. Undersampling does not take profit from a high amount of negative cases, whereas oversampling gives a more precise representation of the negative class than the positive one. For ANN, oversampling is favoured because it is data-greedy. There is no apparent reason why SVM obtained its higher performance on the unbalanced dataset. There is no trend regarding the stratification of the dataset.

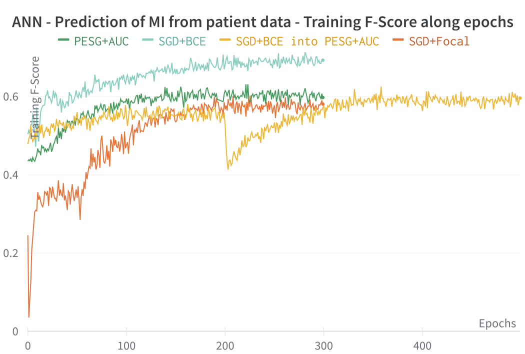


Figure A.1: Evolution of the training F1-Score along epochs using the ANN to predict MI from patient data, presented in figure 4.1.

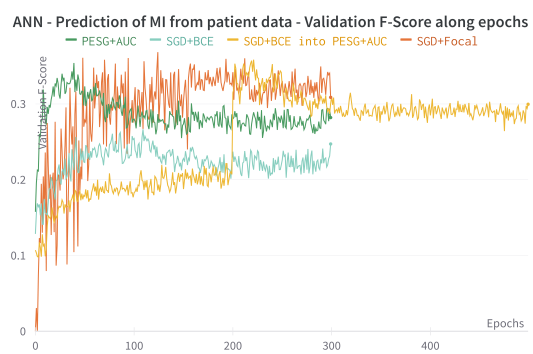


Figure A.2: Evolution of the testing F1-Score along epochs using the ANN to predict MI from patient data, presented in figure 4.1.

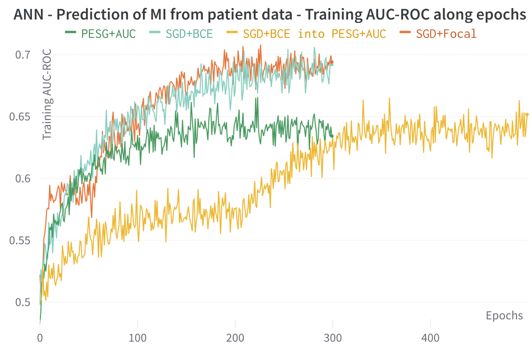


Figure A.3: Evolution of the training AUC along epochs using the ANN to predict MI from patient data, presented in figure 4.1.

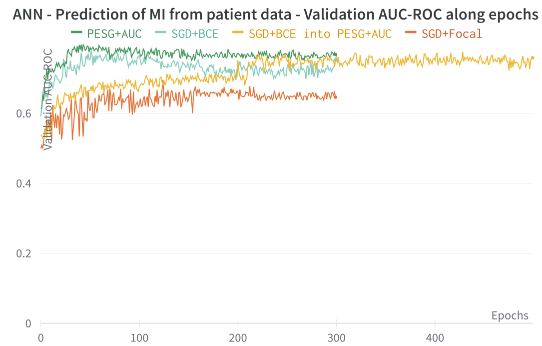


Figure A.4: Evolution of the testing AUC along epochs using the ANN to predict MI from patient data, presented in figure 4.1.

A.3 CNN

A.3.1 Dataset

A.3.1.1 Data preprocessing and extension

All the images have been resized to 1524x1524px; 99.59% of the images had already this dimension. The dataset can be loaded oversampled, undersampled or without a balancing strategy. To help further against imbalance, different data extension methods have been considered (only applied to the training data). They are all implemented with the *transforms* module of *PyTorch*. Each one has a given probability of happening and is applied sequentially in the following order:

1. Random cropping (20% of the time): a subimage is cropped on the image, which can have a ratio between 75% and 125% of the original image and between 80% and 99% of the size of the image. The image is then resized to the input size;
2. Random rotation (20% of the time): the image is rotated between -30° and +30°;
3. Color alteration (20% of the time): brightness is altered between 80% and 120%, contrast between 80% and 120%, saturation between 80% and 120% and hue between -20% and 20%.

For the cropping and rotation, the same values are applied to the raw image and its mask (otherwise, the mask does not provide the correct insight of the sections' position). The values have been fixed heuristically. Gaussian blur is also implemented but not used due to low performance and high computation time.

A.3.1.2 FFCV

Each input (patient) consists of twelves images 1524x1524px which is quite big for the Deep Learning field. One input contains $12 * 1524 * 1524 = 27'870'912$ pixels, which weights 28 Mb if each pixel weights 1 byte (0 to 255 values). With the same computation method, the *complete* MNIST dataset weights 47 Mb and the *complete* CIFAR 184 Mb. A batch of our images is heavier than these whole datasets. The significant size of the input prevents using big batches and infers that most of the training time is spent on data loading and not optimisation. It was not possible to reduce the size of the image because the MI event is a local event.

A special library, FFCV [23], has been used to reduce the loading time. The library optimises the loading and processing pipeline. To do so, our whole dataset had to be saved into an FFCV file (which has the *.beton* file extension). Table A.1 shows the speed performance obtained by different data loading strategies. The first one, "no optimisation" is the naive implementation. Then, some improvement can be made by making sure to use all the dataloaders parameters ("optimisation"). The FFCV strategy is more efficient than the others; it is almost three times faster than the naive one and two times faster than the optimised one. Please note that the first epoch is longer for the optimised and FFCV methods as some setup is done.
Despite the impressive performance of the library, it only implements a subset of the dataloaders functions of *PyTorch*. Specifically, the overbalancing strategy, which consists of giving each batch the same number of negative and positive samples, is not implemented. To compensate for that, a new dataset with repeated positive cases until balance has been created. There is an FFCV file for each dataset sampling strategy. Other drawbacks of the library are that the generated FFCV files are very heavy (87 GB for the oversampled one) and cannot be updated once created.

Table A.1: Duration of an epoch with different optimisation configuration. This example uses a mini-dataset consisting of 28 samples in training, 8 in testing and a batch size of 4.

Configuration	**Epoch**	**Training (s)**	**Validation (s)**	**Total duration (s)**
No optimisation	0	46.5	10.4	56.9
	>0	46.5	10.4	**56.9**
Optimisation	0	39.3	14.1	53.4
	>0	31.3	7.1	**38.4**
FFCV	0	40.6	15.3	55.9
	>0	20.7	2.6	**23.3**

A.3.2 Discussion: follow-up

A.3.2.1 Performance along epochs

The evolution of the training and testing F1-Score along epochs of the CNN "Common" architecture with BCE and AUC loss is shown in figures A.5 and A.6. For the CNN "Max" architecture, the same information is displayed in figure A.7 and A.8, for the BCE, AUC and Focal loss. Finally, for the CNN "Max with patient data", figures A.9 and A.10 plot it, for the vanilla implementation, the implementation with higher weight on the arteries prediction ("High artery loss") and the implementation with pretrained CNN and ANN ("pretrained").

A.3.2.2 Interpretability: negative case

In figure A.11, interpretability methods have been applied to the best CNN on the data of a negative patient.

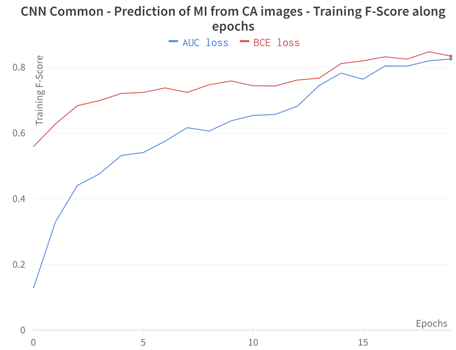


Figure A.5: Training F1-Score along epochs using the CNN common architecture (figure 5.3).

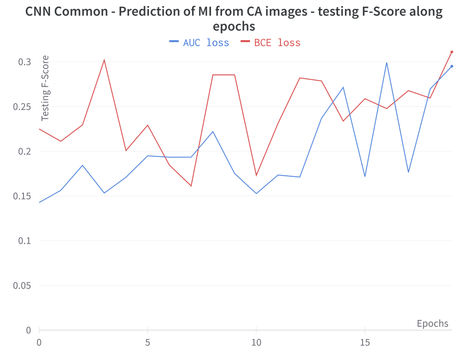


Figure A.6: Testing F1-Score along epochs using the CNN common architecture (figure 5.3).

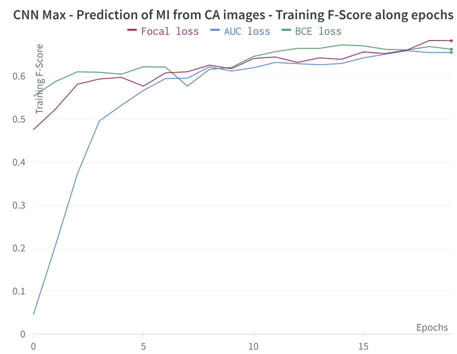


Figure A.7: Training F1-Score along epochs using the CNN max architecture presented in figure 5.4.

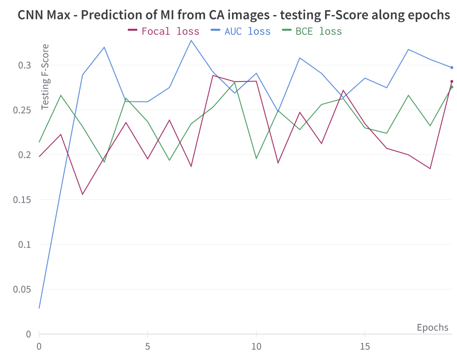


Figure A.8: Testing F1-Score along epochs using the CNN max architecture presented in figure 5.4.

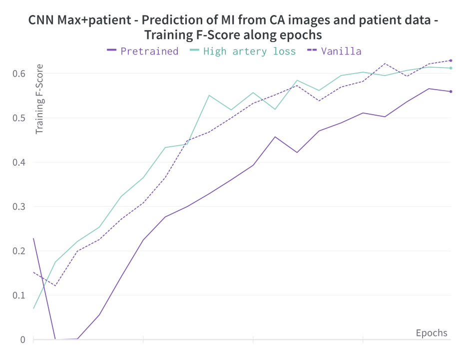


Figure A.9: Training F1-Score along epochs using the CNN max and patient data architecture presented in figure 5.5.

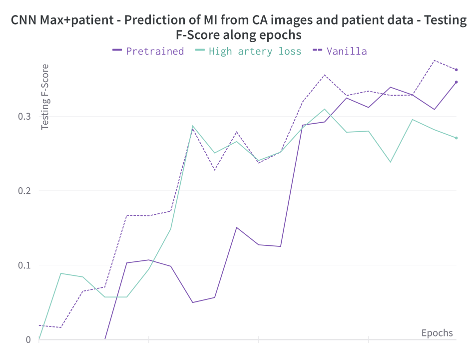


Figure A.10: Testing F1-Score along epochs using the CNN max and patient data architecture presented in figure 5.5.

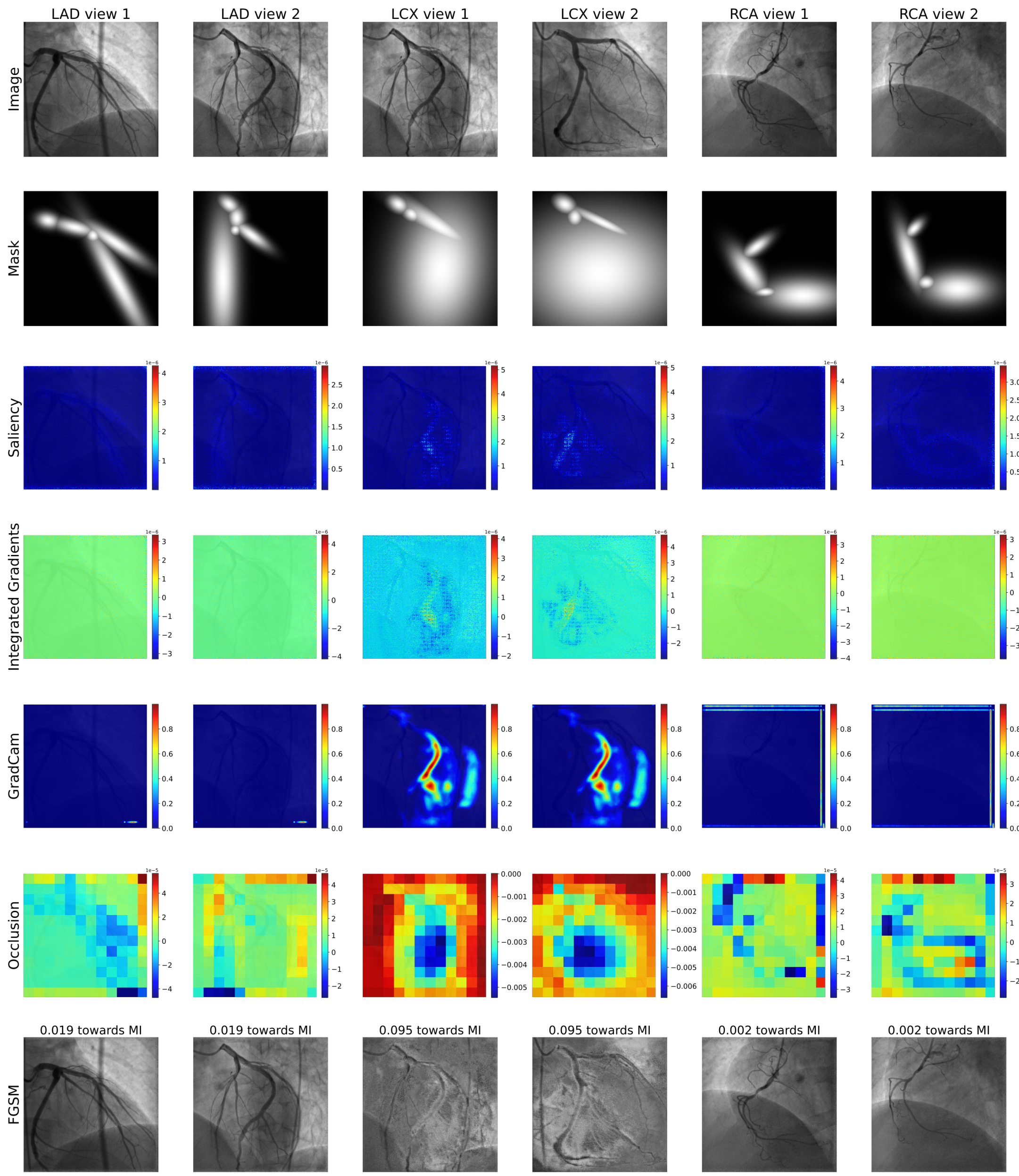


Figure A.11: Interpretability analysis of the Max CNN with patient data architecture 5.5. The input does not contain MI and comes from the testing dataset. The different interpretability methods have been introduced in 3.3.5. The analysis is done artery wise and GradCam returns the same mask for both views.

A.4 Transformers

A.4.1 Patches extraction strategy

The steps of the patch extraction strategy discussed in the thesis are further detailed in the next sections.

A.4.1.1 Box extraction and orientation

From their coordinates, the boxes are extracted. Then, they are rotated with respect to the blood flow (enters from the left of the image and exits on the right). The side from which the blood enters has been found by looking at the two closest sides of two consecutive boxes: for the first box, the closest side will be the side from which the blood exits, and for the second, the one from which it enters. When there is a bifurcation (an artery splits in two), the orientation is computed with respect to the "origin" of the box. The two closest sides are found by looking at the two pairs of nearest corners in the two boxes (without replacement).
Despite all the efforts, the orientation of the boxes is not always guaranteed: the strategy does not work in some particular cases (for example, when the artery makes a loop), and there is some "randomness" (certainly a misunderstanding from my part) on the *CV2* library extraction orientation. It still ensures that most of the time, there is a "blood flow" logic.

A.4.1.2 Centerline computation

Patches could have been extracted randomly on the image. However, arteries are pretty thin, which would have resulted in many background patches. To avoid that, a classical Computer Vision algorithm has been adapted to detect the centerline of the arteries [45]. It consists of filters applied on top of each other (median, Frangi, Otsu thresholding) and standard computer vision tools (skeletonisation, closing). The performance is not outstanding, mainly due to the challenging images (medical tools, sharp background, ...), but it gives a better insight than randomness. The centerline detection is done at the box level and not on the whole image because it gives better results.

A.4.1.3 Preprocessing

The preprocessing is done at the box level before the patch extraction (applying modification on small patches may be too destructive). No resizing has been applied. Different data extension have been considered (implemented with the *transforms* module of *PyTorch*). They are similar to the one applied to the CNN images but less aggressive. They have their own probability and are applied sequentially:

1. Random cropping (10% of the time): a subimage is cropped on the image, which can have an image ratio between 90% and 110% of the original image and between 90% and 99% of the size of the image. The image is then resized to the input size;
2. Gaussian blur (10% of the time): a Gaussian filter is applied to the image; it has a kernel size of 3x3 and a standard deviation between 0.001 and 0.01;
3. Random rotation (10% of the time): the image is rotated between -2° and +2°.

For the cropping and rotation, the same values are applied both on the box and on its detected centerline (otherwise, the centerline does not provide the correct insight). All the values have been fixed heuristically.

A.4.1.4 Patch extraction

As previously explained, the images are oriented so that the blood enters from the left and exits on the right. So, the patches are sampled semi-uniformly from left to right. That defines the x-position (width of the image). The "semi" means a small random noise is added to the x-position. For each defined x-position, the y-position (height) is chosen randomly on the pixels classified as centerline at this x-position. If none of them is, the y-position is selected randomly. There is a tunable probability (fixed to 10%) that the y-position is entirely randomly selected, which allows still to extract information in case of poor centerline detection. In figure 6.1, the x-positions are the red dashed lines, and the red dot indicates the y-positions (and the centre of the patch extraction).
The size of all the extracted patches has been fixed to 64x64px. The arteries are typically 60px wide; thus, this size is insufficient. A bigger size would have been interesting to be sure to contain the whole artery width on the image and see its surroundings, but this parameter significantly influences the network size. The number of patches extracted depends on the box type, as some tend to be bigger than others (see figure 2.5). It has been decided to extract 32 patches for the magenta boxes, 64 for the yellow ones and 128 for the green and brown ones. Our patch extraction method suffers from the poor centre line and the assumption that the artery starts on the left of the image and ends on the right (sometimes, the artery is curvy and ends in the centre of the image).

A.4.1.5 Final list extraction

All the patches have been extracted sequentially (the first one on the list is the one from the left and the last from the right). The list of each section is concatenated following their apparition order (magenta-yellow-green-brown). That results in a sequence of patches from the start to the end of the artery. However, this order does not make sense when there is a bifurcation, as the two boxes that emerge from the bifurcation should be *parallel*. There is no clear solution for a sequential input, but a graphical representation may solve this issue. Finally, each patient has a sequence of 352 ($= 32 + 64 + 128 + 128$) patches with the blood flow logic for each view of each artery. The dataset can be raw, undersampled or oversampled (oversampling in batch).

A.4.2 Performance along epochs

In figure A.12 and A.13 the evolution of the training and testing F1-Score along epochs of the transformer architecture using BCE and Focal loss is presented. In figure A.14 and A.15 the same metrics are presented but for the network optimised on AUC loss. Figures A.16 and A.17 show the F1-Score for the Sequence and Softmax transformers.
The evolution of the F1-Score along epochs (figures A.12 and A.13) shows that BCE needs some epochs before reaching a non-null validation F1-Score. However, it reaches a comparable performance and seems to overfit less than the other (its training F1-Score is lower than the others for the same validation F1-Score).
Both preloading and pretraining approaches have a similar behaviour regarding the evolution of the validation F1-Score (figure A.15). For the training F1-Score (Figure A.12), the situation is different; the pretrained strategy reaches a higher performance (and thus overfits more). In the same picture, it is surprising that the preloaded network achieves poorly at the first epoch, meaning that what was learned is unlearned (which is possible, as it is not using the same loss, so has not the same objective and faces slightly different data).

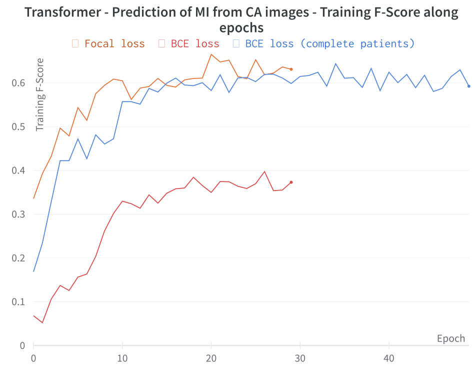


Figure A.12: Training F1-Score along epochs for different learning strategies using the transformer structure optimised with BCE or Focal loss to predict MI from CA, figure 6.4

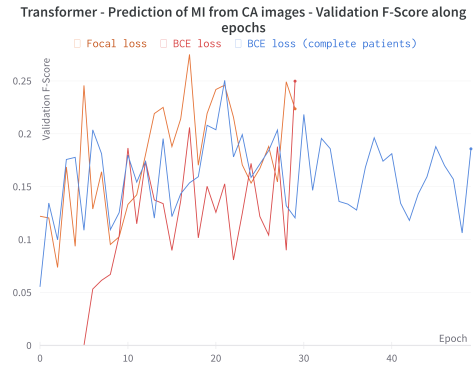


Figure A.13: Testing F1-Score along epochs for different strategies using the transformer structure optimised with BCE or Focal loss to predict MI from CA, figure 6.4

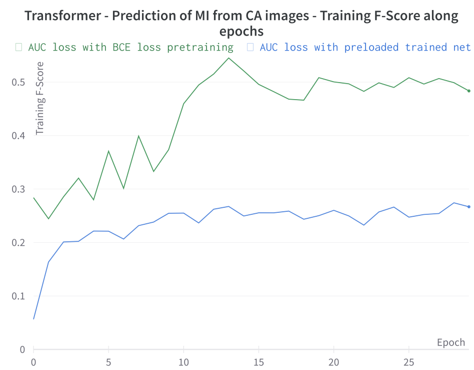


Figure A.14: Training F1-Score along epochs for different strategies using the transformer structure optimised on AUC losses to predict MI from CA, 6.4

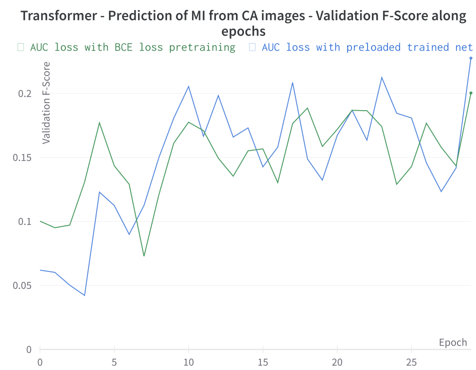


Figure A.15: Testing F1-Score along epochs for different strategies using the transformer structure optimised on AUC losses to predict MI from CA, 6.4

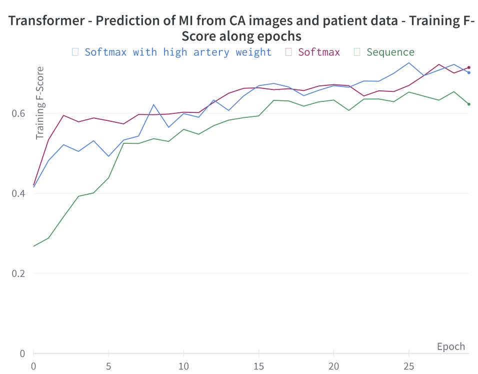


Figure A.16: Training F1-Score along epochs for different strategies using the transformer structure to predict MI from CA and patient data, figure 6.5

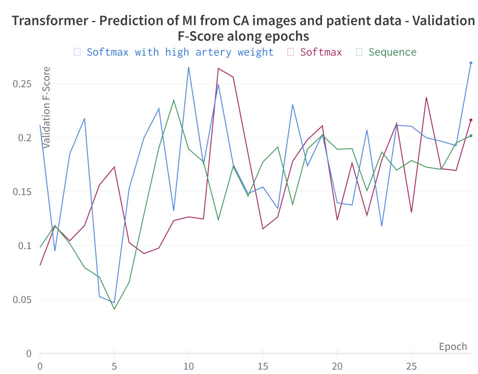


Figure A.17: Testing F1-Score along epochs for different strategies using the transformer structure to predict MI from CA and patient data, figure 6.5