Prediction of myocardial infection risk after stenosis diagnosis

Acquati Francesco, Blackburn Michaël, Ménaësse Audrey
Life Sciences Engineering, EPF Lausanne, Switzerland

Under the supervision of Thanou Dorina, *Chair of Mathematical Data Science, EPF Lausanne, Switzerland*

***Abstract*—Vascular stenosis is a medical condition that can lead to myocardial infarction several years after diagnosis. The purpose of this project is to detect this complication risk on angiograph images.**

I. Introduction

Stenosis is a medical condition consisting in an abnormal narrowing of a hollow organ. In particular, vascular stenosis is a tightening or hardening of a blood vessel which can have various causes.

Vascular stenosis can be detected with a stethoscope thanks to a characteristic sound related to the unusual blood flow. Nevertheless, medical imaging is required to confirm the diagnosis (1). This study is based on catheter angiography images taken at the cardiology department of the University Hospital of Lausanne (CHUV). These images were taken as a primary scan on patients with vascular stenosis symptoms. After a period of 1 to 5 years, some patients experienced myocardial infarction and the stenosis areas responsible of this complications were manually annotated by physicians.

The purpose of this project is to understand if these imaged stenosis areas can be used to predict whether they will cause myocardial infarction and thus to be able to anticipate possible interventions and treatments that could avoid severe consequences. To do so, transfer learning on some popular existing architectures was used and the different results obtained were compared in an attempt to determine the most adequate method for this application.

The pipeline developed to choose and train the best model is presented in the figure 1.

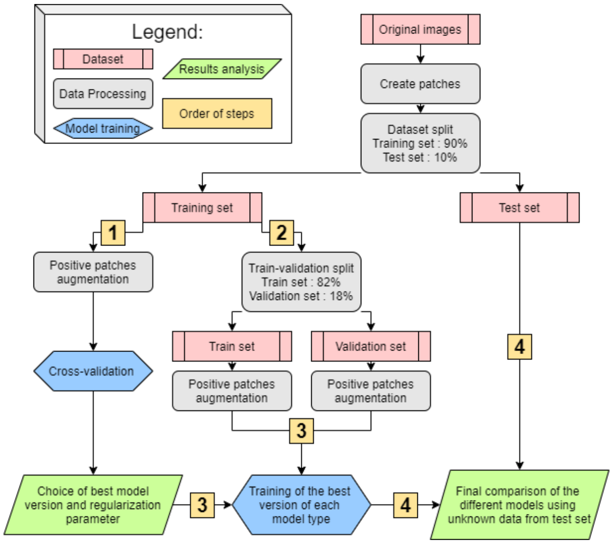

Figure 1. Structure of the project to choose and train a deep learning model to recognize risks of stenosis complications from angiography images

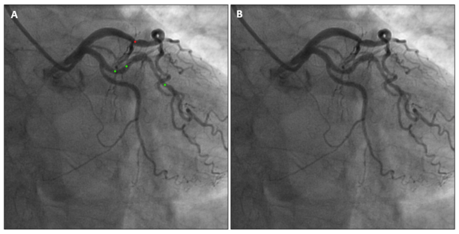

Figure 2. Example of a labelled (A) and raw (B) image

II. Image preprocessing

A. Data structure

The entire dataset is composed of 379 raw and 374 labelled images containing 1014x1014 pixels. Labelled images have been manually annotated by cardiologists from CHUV with colored dots at the location of the stenosis. A green dot means that the stenosis area did not cause myocardial infarction while a red dot means that this stenosis did cause myocardial infarction in a period of 1 to 5 years. For each patient, four images were taken at different camera angles.

B. Selection of regions of interest

As shown in figure 2, the regions of interest are already defined by the position of the physician annotation. The classification task is performed on patches containing one stenosis each. By default the size of these patches is set at 224x224 pixels to match the required input size of the pretrained models of interest. However, to ensure that only one annotated dot is included per patch, the crop size was reduced if another dot was in the vicinity or the border of the image. The result is the largest possible crop size around each label with an upper limit of 224x224 pixels.

The patches were then saved under a Numpy format in a zipped `npz` file that can be easily and quickly read. These files include 2 lists; the images and the labels 0 or 1.

C. Data splitting

The project pipeline described in figure 1 contains two distinct steps of data splitting. After the creation of the patches around the regions of interest, a first split was performed. This division consists in isolating a test set of 10% of the total patches allowing to assess the performance of the different classification methods on completely unknown images. The splitting of the patches was stratified to have the same proportion of positive and negative labels in both the sets. We now have a test set and a training set. For the cross validation step, the training set obtained here was kept as it is, as the cross-validation procedure already includes the isolation of a data fold for validation. After determining the best models through cross validation, this training set was further split to extract a validation set.

On the other hand, this second split is necessary before a regular model training. That is why, the step number two from the figure 1 includes a data splitting. The ratio between the train set and the validation set is 82% / 18% so that the final repartition of the data is 70% for train set, 20% for validation set and 10% for the test set.

D. Data augmentation

The data set provided for this project is composed of 372 pairs of raw and annotated images from which 713 patches can be extracted around regions of interest. However, the distribution between positive and negative patches is highly unbalanced as only 28.6% of the patches are positive.

To avoid bias in the model training due to this unbalance, the number of positive patches was augmented using image duplication. We justify this step as every time a patch is loaded from a data `npz` file, it undergoes a series of random transformations, reading two identical patches will provide two different transformed images. The patches were augmented upon sampling during the training and testing phases.

The transformations were implemented using the *albumentations* library as the random transform sequences can be reproduced by setting the random seeds. It also includes many tuneable and diverse transformations. The transformations used are as follows:

align=parleft

- MedianBlur(blur_limit=3, p=0.3)
- Rotate(limit=45, interpolation=1, border_mode=4, p=0.75)
- ShiftScaleRotate(shift_limit=0.1, scale_limit=0, rotate_limit=0, interpolation=1, border_mode=4, p=0.75)
- Resize(224, 224)

III. Classification methods

A. Deeplearning Models

Models pretrained on the CIFAR-10 dataset from Pytorch's TorchVision module were used as starting points. Specifically, three variations of DenseNets, ResNets and VGGs were each trained and compared. The pretrained models by default output 1000 classes. For our application, we reshaped the final output layer to two cases. In all cases, all layers were trained. No layers were frozen.

1) Training Procedure: The models were trained using a stochastic gradient descent optimizer and learning rate of 1e-2. A Pytorch dataloader was used to load the data in mini-batches of size 10. The images loaded with the dataloader were passed through a dataset class which applied the specified transformations and normalization upon sampling. Normalization was done as required by Pytorch's pretrained models with mean = [0.485, 0.456, 0.406] and std = [0.229, 0.224, 0.225]. As such, the transformed images are never saved, they are created at each mini-batch only. We decided to proceed this way instead of creating and saving all of the transformations beforehand to save storage space. Additionally, by applying random transforms upon sampling, there is no limit to how many transformed data we use as each epoch generates a new set. A standard cross entropy loss function was used for the training.

2) Evaluation Procedure: At each training epoch, the validation accuracy is calculated. The validation images are also augmented through the same transforms as the training images. This is done in an attempt to increase not only the amount of train data but also validation data. In addition to the accuracy, we also note the specificity, sensitivity and F1-score of the validation set.

3) Model Selection: In order to narrow down the list of potential models, a 5-fold cross-validation was performed on the 3 variations of the 3 families of models with 3 different values of weight decay regularization determined empirically: 0, $1 \cdot 10^{-5}$ and $1 \cdot 10^{-3}$. Box plots were used to visualize the results and they can be seen in the subsequent sections. To conduct the cross-validation, the initial train set (90% of total data) was used. As the red data labels of this train set have been augmented through duplication, we suspect that duplicates were present in both the folds used for training and the validation folds. The very high validation accuracy for the CV tests clearly shows this. Although there are no exact duplicates as we apply random transforms to each image, we expect that the transforms of duplicates will be close enough that the model easily detects them. Thus giving a false impression of high sensitivity in CV results.

4) DenseNet: Dense convolutional networks, also called DenseNet, are convolutional networks where each layer is not only connected to the next one, but to all the layers that follow it (2). This way, for a network that contains L layers, a DenseNet will contain $\frac{L(L+1)}{2}$ connections where a traditional convolutional networks will have L connections. This structure has the advantage of reinforcing the features transmission through the layers, reduces the number of parameters of the model and alleviate the vanishing gradient problem. The structure of DenseNet considered in this project are DenseNet121, DenseNet161 and DenseNet201. The results of the validation accuracy obtained with the cross-validation are shown in the figure 5 below.

The configuration providing the best compromise between higher median accuracy and variance is the DenseNet 201 with a regularization parameter of $1 \cdot 10^{-3}$. Thus, this model was selected to be trained on the train and validation datasets which have been split before the positive data augmentation to avoid

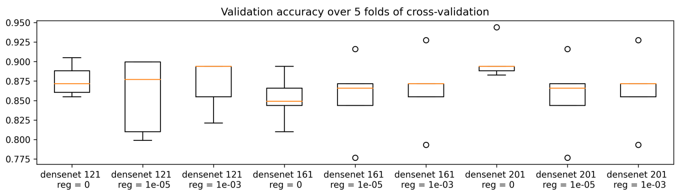

Figure 3. Boxplot of validation accuracy obtained for different DenseNet architectures trained with different weight decay regularization values

any redundancies.

5) ResNet: Residual Neural networks are learning residual functions from the layer inputs (3). The three structures retained are ResNet18, ResNet101 and ResNet152. The best candidate for ResNet was the ResNet 152 with a regularization of 0.

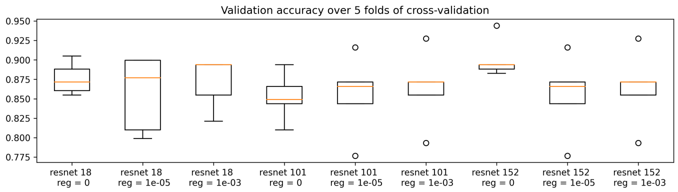

Figure 4. Boxplot of validation accuracy obtained for different ResNet architectures trained with different weight decay regularization values

6) VGG: VGG are very deep convolutional neural networks, thus containing much more parameters than the other architectures (4). Only variations of VGG with batch normalization were used. The structures were vgg11_bn, vgg13_bn and vgg19_bn.

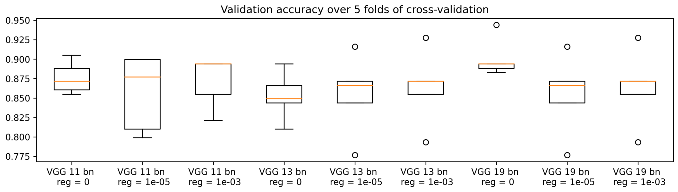

Figure 5. Boxplot of validation accuracy obtained for different VGG architectures trained with different weight decay regularization values

The best VGG model selected to be compared with the other architectures was VGG 19 with batch normalization and a regularization of zero.

B. SIFT detection and Support Vector Machine

To compare the results with a more simple architecture, scale-invariant feature transform (SIFT) was applied to the raw images. A varying number of key-points from 1 to 20 was considered and their descriptors were used to train models. To avoid overfitting, principal components analysis (PCA) was considered and only limited explained variance was included. A linear Support Vector Machine (SVM) with a regularization parameter of 10−4 was applied to classify the images. A KNN with 1 neighbor only and the Minkowski distance was also considered as a possible algorithm instead of SVM. The models were applied on the same 90-10 split used for the final evaluation of the deep learning models.

Model	PCA (exp. var.)	Mean train accuracy (SD)	Mean test accuracy (SD)
SVM	-	0.98 (0.04)	0.56 (0.11)
SVM	Yes (90 %)	0.93 (0.11)	0.53 (0.07)
SVM	Yes (80 %)	0.88 (0.11)	0.60 (0.06)
KNN	-	1.00 (0.00)	0.54 (0.09)
KNN	Yes (80 %)	1.00 (0.00)	0.58 (0.11)
KNN	Yes (70 %)	1.00 (0.00)	0.58 (0.10)
KNN	Yes (60 %)	1.00 (0.00)	0.55 (0.05)

Table I
AVERAGE ACCURACY RESULTS OF THE SIMPLE MODELS WITH AN INCREASING AMOUNT OF SIFT KEYPOINTS CONSIDERED FROM 1 TO 20.

IV. RESULTS

Considering the SIFT application, the use of an increasing amount of keypoints lead to an decreasing inclusion of patches. Indeed, the amount of keypoints that can be extracted from an image is limited and depends on the image itself. If the required amount of keypoints was not reached, the patch was discarded. Average results along an increasing amount of keypoints for the use of SVM and KNN both with and without PCA are reported in table I. Accuracy results on the train and test set with increasing amount of keypoints can be seen in figure 6 for the SVM model applying PCA with 80 % explained variance

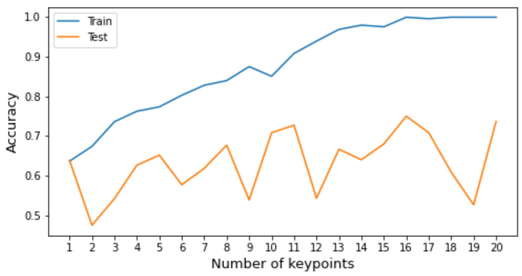

Figure 6. Evolution of training and test accuracy with an increasing amount of keypoints for the SVM model, applying PCA with an explained variance of 80 %.

If the accuracy results obtained with the cross-validation are very encouraging, they cannot be used to draw conclusions are they are biased by the way the negative data has been augmented before the fold splitting for cross-validation.

As expected, one can notice on figure 7 that when the validation of the model is performed with a dataset completely independent from the training dataset, the validation accuracy drops to around 67% contrary to 90% during the cross-validation.

Another important element to notice is that the training accuracy is converging towards 100 %. This means that the model is overfitting on the training data. In theory this phenomenon should be avoided, however this was expected as all three models considered here have more than a million parameters and are trained on a total of 975 patches of 224x224 pixels

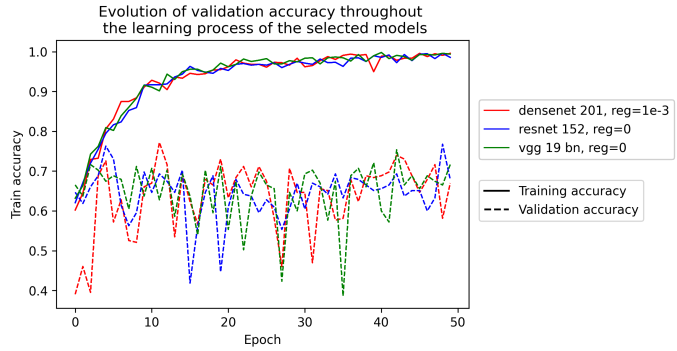

Figure 7. Evolution of training and validation accuracy throughout the learning process of the three different model architectures

after data augmentation. This phenomenon can be limited by increasing or variability of the transforms applied to the images or considering larger datasets.

V. Discussion

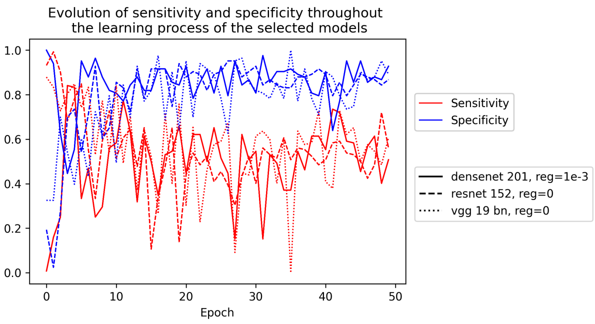

Figure 8. Evolution of sensitivity and specificity throughout the learning process of the three different model architectures

As shown in figure 8, for all the DenseNet, ResNet or VGG models considered in this project, the sensitivity and specificity have mirror behaviors. This is expected as the networks are performing binary classification, so if the classifier is more strict in the detection of events, more events will remain undetected.

For the sake of the patients, it is necessary to seek high sensitivity rather than high specificity. Nevertheless, the specificity obtained during the testing phases of the models approaches one, while the sensitivity hardly exceeds 0.6.

This can be explained by the unbalanced original dataset. Indeed, to compensate for the lack of positive patches compared to the negative ones, positive patches have been augmented by a factor three using isotropic transforms. This implies a much lower variability inside the positive patches than the negative patches and explains the low sensitivity of the model on unknown images.

Furthermore, one can notice that during the training of the model, the specificity increases while the sensitivity decreases. That is why it is interesting to wonder to what extent changing the paradigm of the problem can help improve this issue. Indeed, if all the patients are considered at risk by default and the event to be detected is the absence of future medical complications, the increasing specificity can then correspond to the goal of this classification task.

Results of the simple models applied on SIFTs show how the SVM better performs compared to KNN. Indeed, even when applying PCA, the performance of the KNN is perfect on the train set and random on the test set. On the other hand, the use of PCA before applying the SVM model effectively increases its performance on the test set. The large variability that is obtained both in train and testing performances could be due to the varying ratio between the train and the test sizes. Indeed, even if they are initially chosen to be 90 % and 10 % of the original dataset, the use of SIFT leads to discarding patches if they do not have a sufficient amount of keypoints. For instance, when considering 20 keypoints, the original dataset of 975 patches for the train and 135 for the test set reduces to a dataset with 209 patches for the train and only 19 for the test set.

VI. Summary

Our experiments have not led to a fully successful model. The problem we are trying to solve is novel and, to our knowledge, has not been attempted yet. Recent research has successly looked into the detection of a stenosis with CNNs (5), but none have attempted to predict the long-term prognostic of these stenoses.

This project has been performed in a short period of time, with limited computing resources and a very limited and unbalanced dataset. The results obtained with the training of the different model architectures cannot confirm or deny that convolutional neural networks are the best tools to accomplish this classification task. Nevertheless, several encouraging conclusions can be made. Indeed, the very high specificity combined with a sensitivity approaching 60% illustrates that the images contain enough information to distinguish the negative patches from the others. Thus, there is good hope that increasing the number of positive patches in the training dataset will improve their detection and so the global performance of the models.

References

[1] “Vascular stenosis.”

[2] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” *arXiv:1608.06993 [cs]*, Jan. 2018. arXiv: 1608.06993.

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *arXiv:1512.03385 [cs]*, Dec. 2015. arXiv: 1512.03385.

[4] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv:1409.1556 [cs]*, Apr. 2015. arXiv: 1409.1556.

[5] J. H. Moon, D. Y. Lee, W. C. Cha, M. J. Chung, K.-S. Lee, B. H. Cho, and J. H. Choi, “Automatic stenosis recognition from coronary angiography using convolutional neural networks,” *Computer Methods and Programs in Biomedicine*, vol. 198, p. 105819, Jan. 2021.

Model architecture	**Number of layers**	**Regularization parameter**	**Accuracy**	**Sensitivity**	**Specificity**	**F1 score**
DenseNet	201	10^{-3}	0.44	0.14	0.92	0.24
ResNet	152	0	0.46	0.24	0.82	0.35
VGG	19	0	0.53	0.38	0.76	0.5
SVM (80% PCA, 10 kp)	-	10^{-4}	0.60	0.72	0.38	0.71

Table II

FINAL RESULTS OF THE DEEP LEARNING NETWORKS COMPARED TO THE SIFT SVM USING PCA WITH 80% EXPLAINED VARIANCE AND A TOTAL OF 10 KEYPOINTS INCLUDED