Applied Research

Timothée Darmaillacq

April 2025

1 Introduction

For the Applied Research, I wanted to work on application of machine learning on medical images. My parents being both Doctors I have grown up hearing about this field, it made sense to me explore the crossing of my studies engineering and medicine and machine learning stand out to me. I had the opportunity to work for a cardiologist during the summer, which further pushed me to explore this path.

After deep search around Machine Learning for Echocardiographic Applications I founded and chose this one :
Deep Learning for Segmentation using an Open Large-Scale Dataset in 2D Echocardiography [1]

2 Summary

This paper, published in August 2019 in the IEEE Transactions on Medical Imaging (TMI), a highly regarded journal with an H-index of 259. Made from collaboration between institutions in France, Norway, Belgium, and Canada, supported by major national research programs from both France and Norway.

This paper aims to evaluate the effectiveness of state-of-the-art encoder-decoder deep learning models in segmenting cardiac structures and estimating clinical indices from 2D echocardiographic images. To support this, the authors introduce the CAMUS dataset, at the time the largest publicly available, fully annotated echocardiography dataset and use it to benchmark model performance against expert cardiologist assessments.

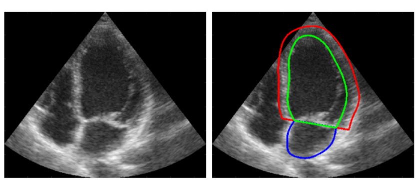
The study demonstrated that encoder-decoder networks, particularly U-Net, achieved highly accurate and robust segmentation of 2D echocardiographic images, with performance close to expert annotations. While U-Net outperformed more complex architectures and showed strong generalization even with varied image quality, results still fell slightly short of intra-observer consistency, highlighting opportunities for further improvement.

This paper is detailed and well-structured, spanning 13 pages and including an appendix that elaborates on specific points. Combined with the accessible, open-source dataset provided, it gives me confidence in the feasibility of re implementing the approach, even for the challenging task of training a CNN model.

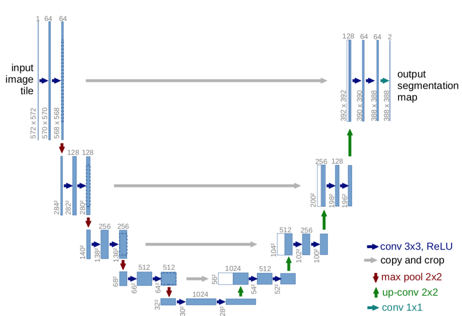
3 Re implementation

I attempted to reimplement one of the CNN U-Net models from this paper, designed for heart mass segmentation on 2D echocardiographic images, specifically the version optimized for training speed, which yielded good results but is faster to train.

A CNN (Convolutional Neural Network) is a deep learning architecture designed for processing structured grid data like images. U-Net[2] is a deep learning architecture designed for image segmentation, particularly in medical imaging, using an encoder-decoder structure with skip connections that help retain spatial details lost during downsampling while stabilizing gradients. The network follows a specific convolutional scheme with two 3x3 convolutional layers at each downsampling and upsampling step, doubling the number of features during downsampling and halving them during upsampling, and is often optimized for different applications to achieve the best segmentation performance.



(a) Example of echocardiographic image and its segmentation.



(b) U-Net Architecture Ronneberger 2015[2]

TABLE I

U-NET 1 ARCHITECTURE

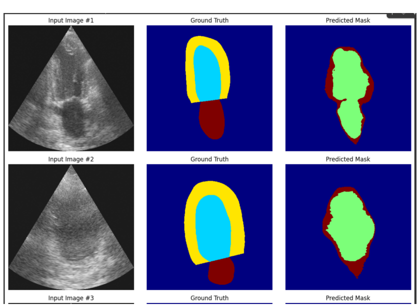
Level	Layer	Kernel / Pool size	Activation	Connection
D1	Conv Conv MaxPooling	32 (3,3) 32 (3,3) (2*2)	ReLU ReLU	*
D2	Conv Conv MaxPooling	32 (3,3) 32 (3,3) (2*2)	ReLU ReLU	**
D3	Conv Conv MaxPooling	64 (3,3) 64 (3,3) (2*2)	ReLU ReLU	***
D4	Conv Conv MaxPooling	128 (3,3) 128 (3,3) (2*2)	ReLU ReLU	****
D5	Conv Conv MaxPooling	128 (3,3) 128 (3,3) (2*2)	ReLU ReLU	*****
D6	Conv Conv	128 (3,3) 128 (3,3)	ReLU ReLU	
U1	UpSampling Conv Conv	(2,2) 128 (3,3) 128 (3,3)	ReLU ReLU	*****
U2	UpSampling Conv Conv	(2,2) 128 (3,3) 128 (3,3)	ReLU ReLU	****
U3	UpSampling Conv Conv	(2,2) 64 (3,3) 64 (3,3)	ReLU ReLU	***
U4	UpSampling Conv Conv	(2,2) 32 (3,3) 32 (3,3)	ReLU ReLU	**
U5	UpSampling Conv Conv	(2,2) 16 (3,3) 16 (3,3)	ReLU ReLU	*
Seg	Conv	4 (1,1)	Softmax	

(c) U-Net 1 layers of the paper[1]

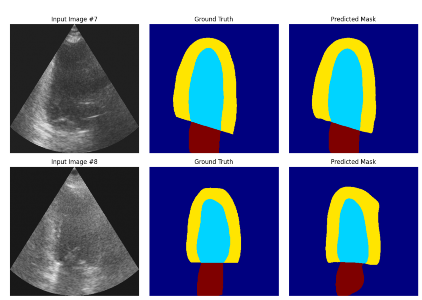
Figure 1: Detail on U-Net Architectures

When tackling a task like this, one of the main challenges is to have sufficient computational power. Since my PC doesn't have a GPU, I first tried using the IFT server. Unfortunately, I ran into issues because I could not get TensorFlow, one of the most common frameworks for machine learning, set up properly. To keep things moving, I switched to Google Colab, a cloud-based platform, and coded in Python using Jupyter notebook. The first thing I did was download the dataset and load it into the coding environment.
Next, I spent some time exploring the dataset to make sure I understood its structure and could access the necessary elements. Once I was familiar with it, I focused on the preprocessing step, which helps format the data, reduce bias, and ultimately improve the model's performance. Initially, I planned to preprocess the data manually, but during my research, I found that a preprocessed version of the dataset was available, so I decided to use that to give myself a better chance of success. Some of the key preprocessing steps included Z-score normalization and contrast stretching, which adjust the intensity range of the images to a specific scale of [-0.2, 1.2], which a counter-intuitive process used solely in this kind of medical image processing.

I replicated the model architecture described in the paper, along with their evaluation method using the Dice coefficient. The model takes as input a 384x384 grayscale image and outputs a segmentation mask of the same size, with 4 channels representing the different segmentation classes. To validate my implementation, I first ran a short test training to ensure that everything worked correctly. This initial run used 250 images over 4 epochs and took 36 minutes to complete. Although the results were not optimal, they were consistent and showed that the process was working as expected. Based on the guidance provided in the paper, I then proceeded with a more substantial training session, still using 250 frames as recommended, but this time extending it over 100 epochs. This longer training took approximately 8 hours and 45 minutes and produced promising results. I evaluated the model's performance using the Dice index, and my model achieved a score of 0.9228, compared to 0.934 reported in the paper. Given how close the results are, I consider this a success.



(a) First model prediction examples



(b) Final model prediction examples

Throughout this project, I learned a lot, starting with the research process, how to find relevant papers, evaluate their credibility, and assess their reproducibility. I also gained significant experience on the technical side of CNNs. It was my first time training a model from start to finish, and I learned how to use tools like Python, Jupyter Notebooks, and TensorFlow.

Building on an existing, well-documented model made the whole process achievable and helped me better understand overall the complexity of this kind of software, as well as each essential step. In the end, implementing a U-Net-based segmentation model was more accessible than it first appeared. While fine-tuning and optimization definitely involve many subtleties, starting with a solid foundation led to good results. My model's performance was close to that reported in the reference paper, showing that even complex deep learning tools aren't completely out of reach.

References

[1] Sarah Leclerc, Erik Smistad, Joao Pedrosa, Andreas Ostvik, Frederic Cervenansky, Florian Espinosa, Torvald Espeland, Erik Andreas Rye Berg, Pierre-Marc Jodoin, Thomas Grenier, Carole Lartizien, Jan Dhooge, Lasse Lovstakken, and Olivier Bernard. Deep learning for segmentation using an open large-scale dataset in 2d echocardiography. *IEEE Transactions on Medical Imaging*, 38(9):2198–2210, September 2019.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *in Proc. MICCAI*, pages 234–241, 2015.