A Combined Deep-Learning and Deformable-Model Approach to Fully Automatic Segmentation of the Left Ventricle in Cardiac MRI

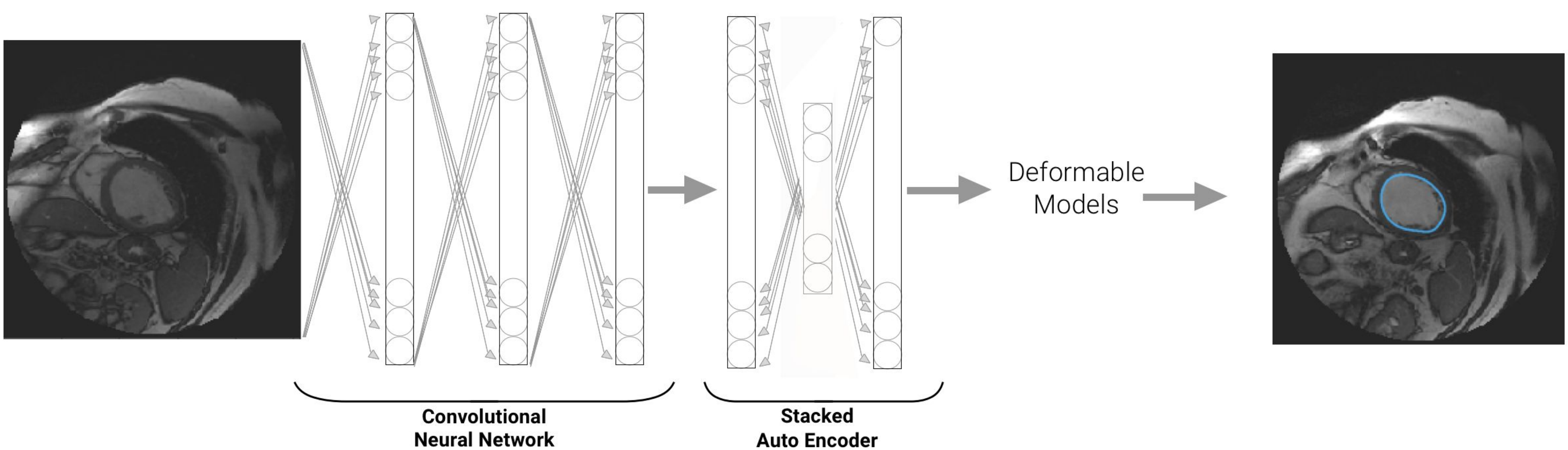
Sharone Dayan

Alexandre Attia

Introduction

Objective :
Automatically segment the Left Ventricle in cardiac MRI

Approach :
Convolutional Neural Network + Stacked AutoEncoder + Deformable Models



Dataset explanations

Dataset

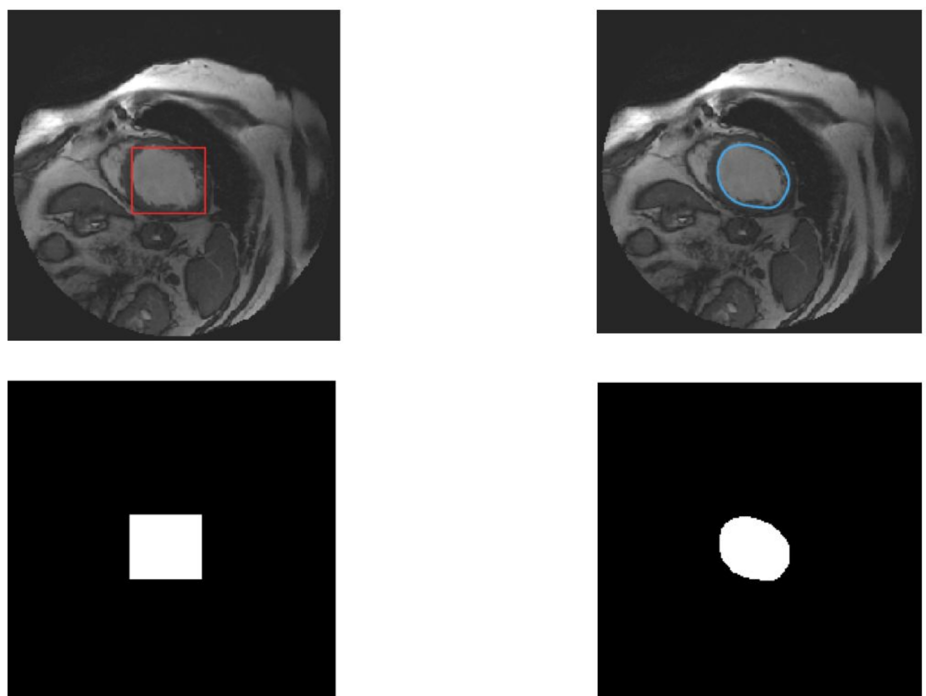
MICCAI 2009 challenge database :
Left Ventricle MR images + manually delineated contours
Trainset : 495 (MRI/contours)

Notations

Region of Interest — Contour shape

with MR image

binary mask

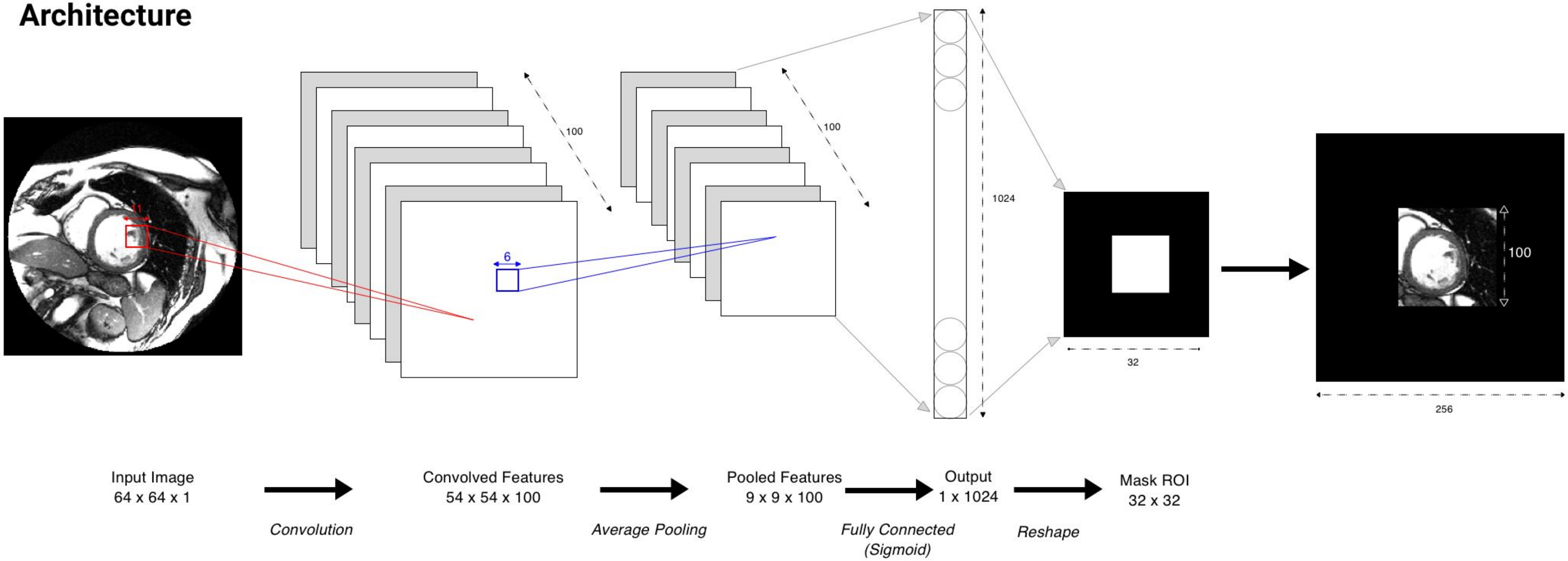


Convolutional Neural Network to locate the Left Ventricle

Main Idea

Generate a Region of Interest (ROI) around the Left Ventricle
Input : Resized Magnetic Resonnace image $(256 \times 256 \rightarrow 64 \times 64)$
Output : Binary mask (100×100)
Initialization : Sparse AE to prevent the lack of data

Architecture



To compute spatial size of the convolution output :

W_2 : output volume size(54)
W_1 : input volume size(64)
F : filter size(11)
P : padding(0)
S : stride(1)

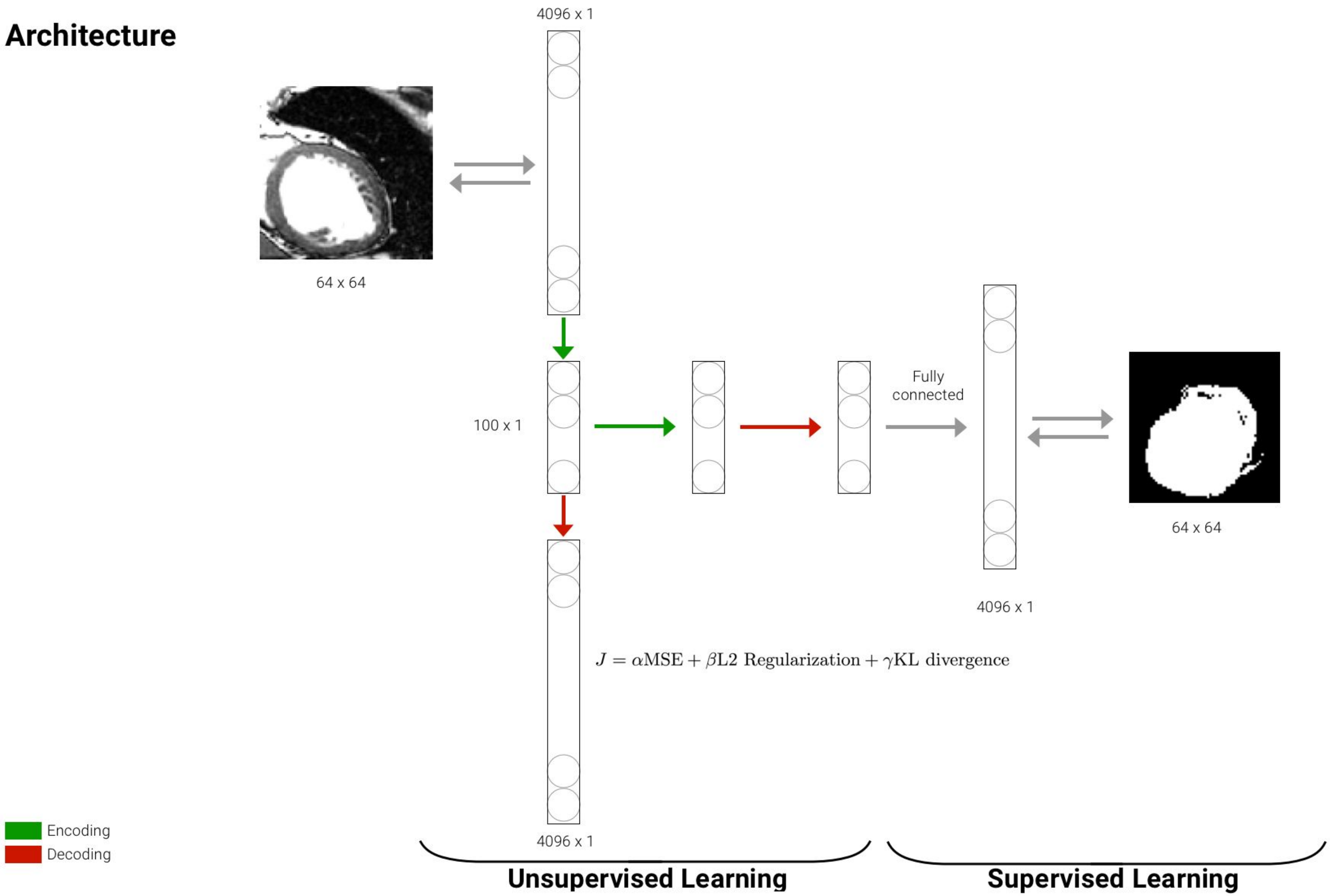
$$W_2 = \frac{W_1 - F + 2P}{S} + 1$$

Stacked AutoEncoder for the Inferred Shape

Main Idea

Segmentation of the Left Ventricle
Input : ROI computed by the CNN
Output : Inferred shape Binary mask

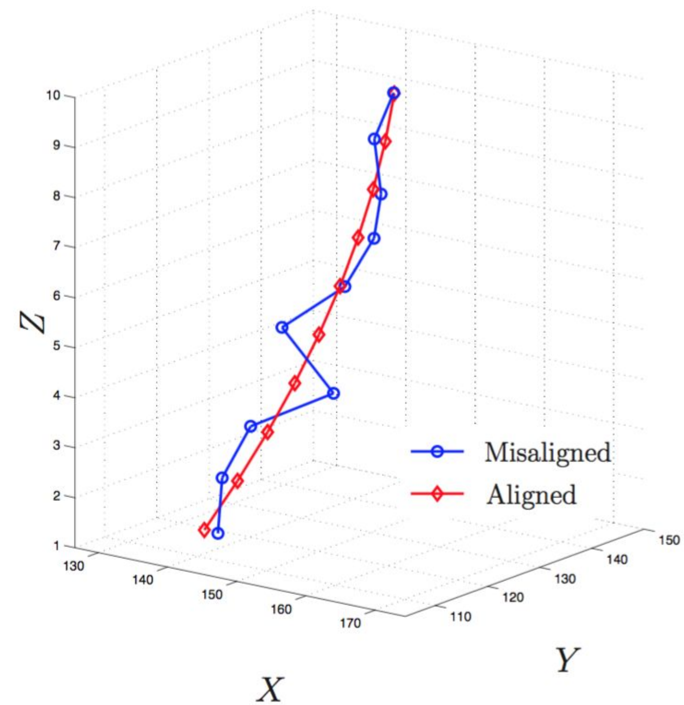
Architecture



Geometric models

$$min_{\phi}\ \underbrace{\alpha_1 E_{len}(\phi)}_{\text{Internal forces}} + \underbrace{\alpha_2 E_{reg}(\phi)}_{\text{External forces}} + \underbrace{\alpha_3 E_{shape}(\phi)}_{\text{Regularization}}$$

Affine registration of the center coordinates



Metrics and evaluation

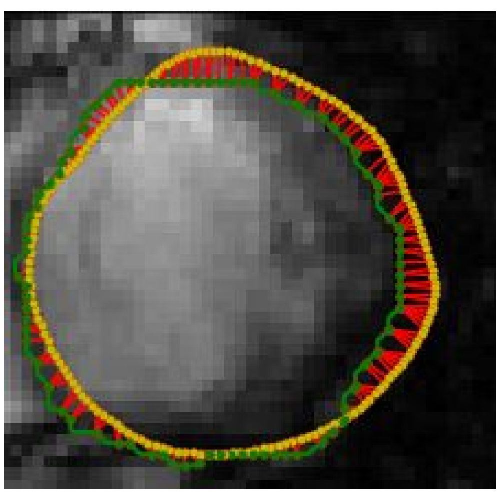
$$DM = \frac{2|A_a \bigcap A_m|}{|A_a| + |A_m|}$$

Dice metric:
Measure of contour overlap between automatically and manually segmentation

$$CC = \frac{3DM - 2}{DM}$$

Conformity coefficient:
Ratio of mis-segmented pixels to the number of correctly segmented pixels

$$APD = \frac{1}{N} \sum_i \|x_i - p(x_i)\|_2$$



Average Perpendicular Distance:
Distance from the predicted contour to the manually drawn, average over all contour points

$p(x) =$ projection of x on C_1
$C_1 = \{x_i, i \in [1, ..., N]\}$

Implementation

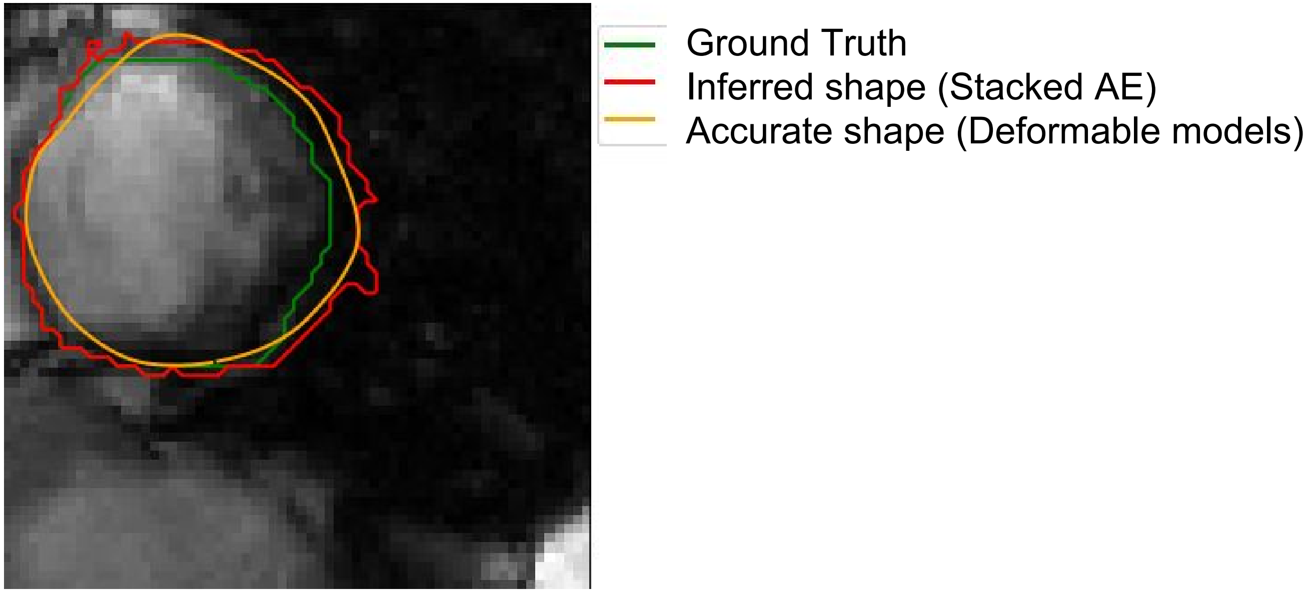
Tools (in Python)

- Keras (TensorFlow backend)
- OpenCV

We have implemented :

- Convolutional Neural Network (without the sparse Auto-Encoder initialization)
- Stacked Auto-Encoder
- Metrics (Dice Metric and Conformity Coefficient)
- Active contours model

Prediction on the validation dataset :

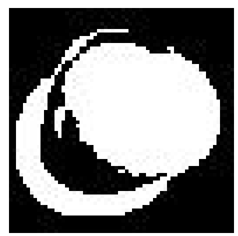
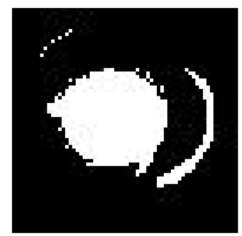
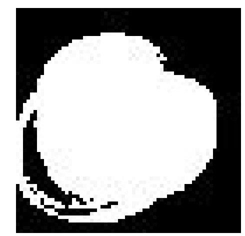
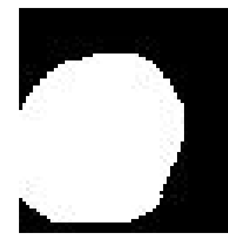


Experiments

Variations of CNN parameters

Possible improvements :

- Deeper neural network *(one more convolutional layer)*
- Wider neural network *(more filters, 200 filters instead of 100 for the convolutional layer)*
- Change convolutions activation (ReLu)
- Replace Average Pooling by Max Pooling

	Simple	Deeper	Larger	MaxPooling	ReLu	Ground Truth
						
mDM	52 %	55 %	48 %	47 %	48 %	

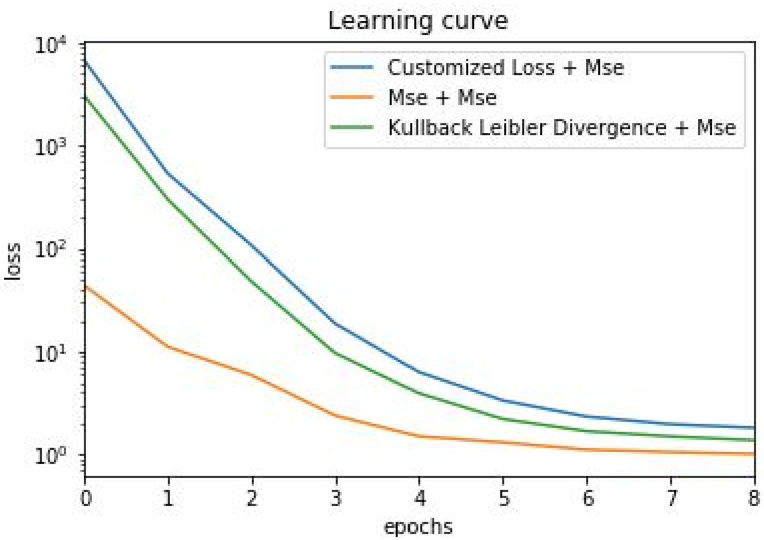
Only a deeper neural network improves slightly the performance

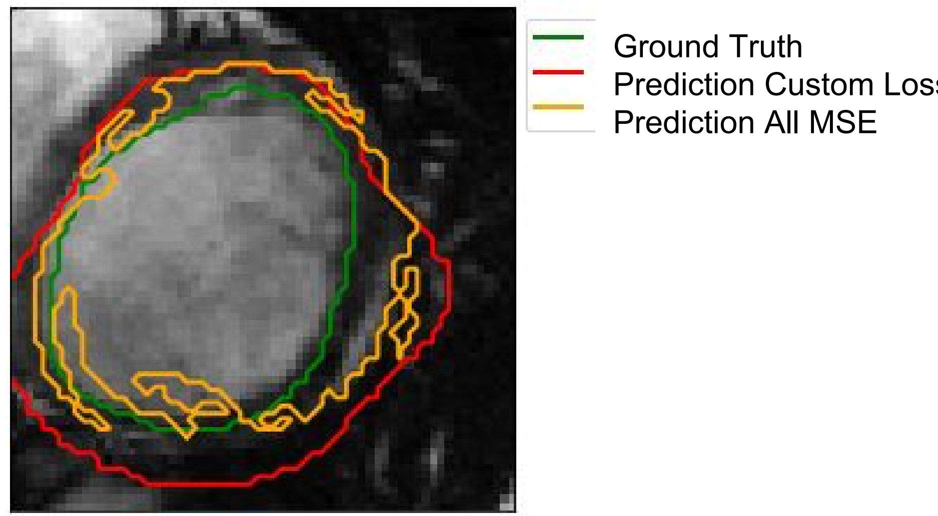
Experiments

Variations of Stacked Auto Encoder parameters

Possible improvements :

- Different initialization for the kernels *(zero weights, random)*
- Modified losses (*MSE, KL, customized loss)*



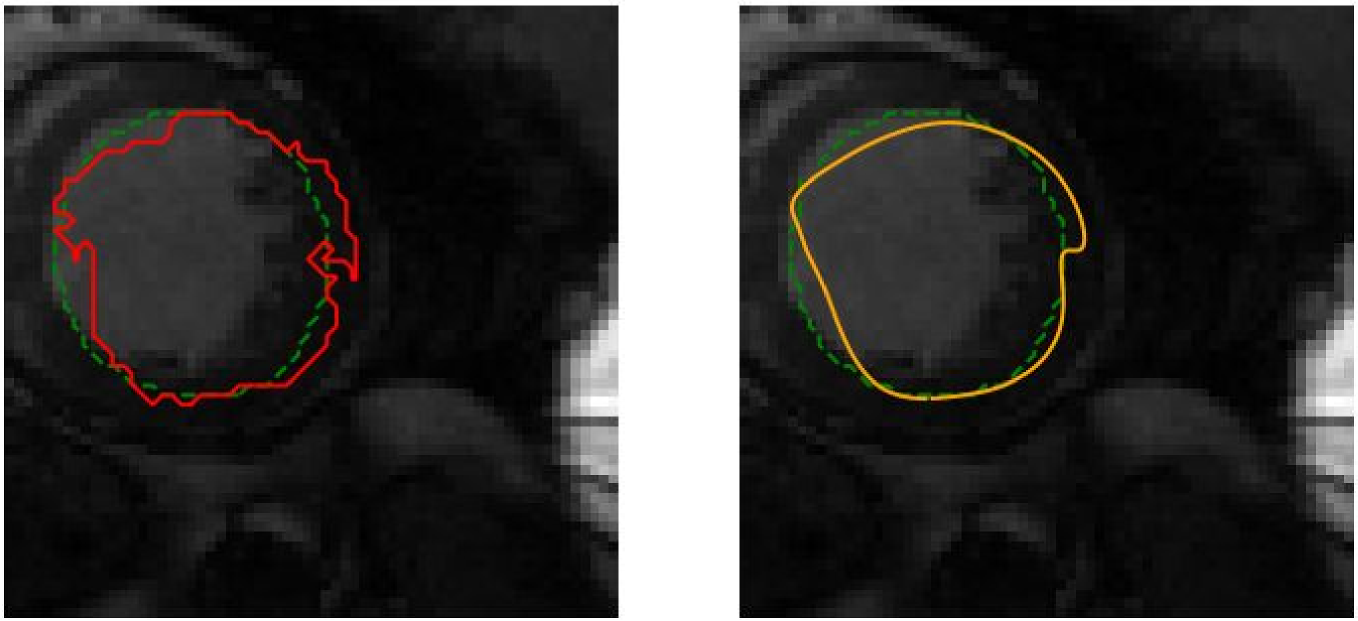


	Original	All MSE	KL + MSE
Dice Metric	31.8%	40.0%	23.4%
Conformity Coefficient	-3.27	-2	-5.54

Experiments

Deformable models

Implementation :
Snakes : Active Contour Model → minimising an energy to increase smoothness and reduce length



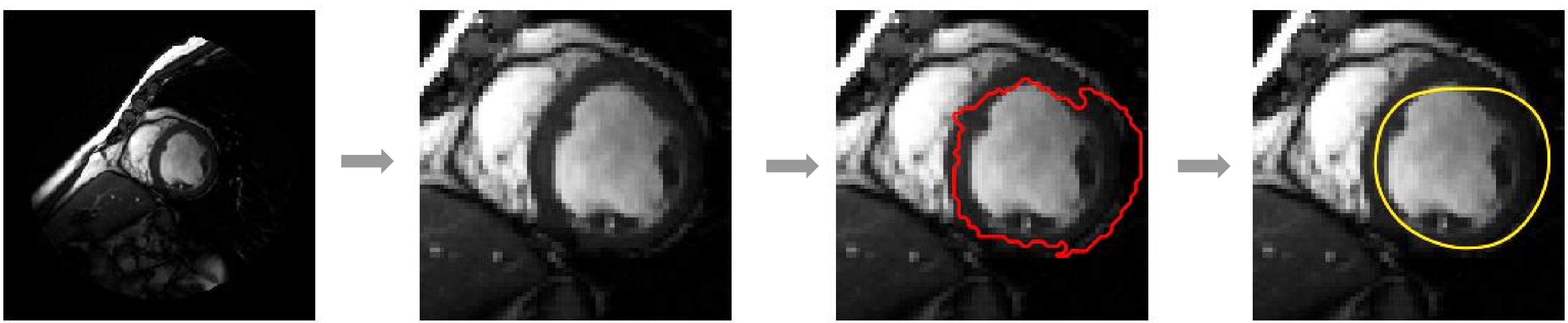
For this picture :
DM without Active Contours : 89 %
DM with Active Contours : 92 %

Results

MRI set :
Online dataset 534 (MRI/contours)
Training dataset 495 (MRI/contours)

Performance on this dataset using CNN (without pre-training) and stacked AE :

	Train set	Validation set	Paper performance	State of the art
Dice Metric	54.8 %	51.6 %	94 %	90 %
Conformity Coefficient	- 1.06	-1.21	1.81	1.76



MR Input image

ROI
CNN Output

Inferred Shape
Stacked AE Output

Final Shape
Active Contours Output