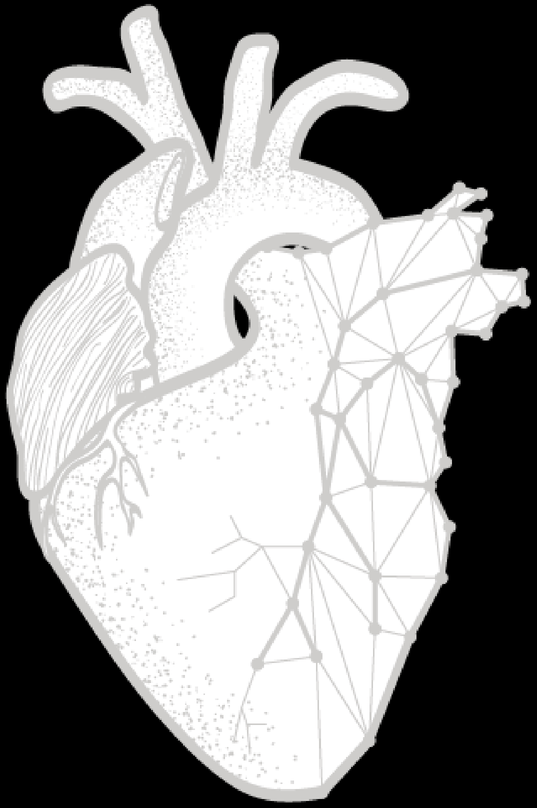


UNIVERSITY
OF TWENTE.

CARDIAC CINE MR SEGMENTATION

ISMRM MACHINE LEARNING TUTORIAL

JELMER WOLTERINK, PHD
DEPARTMENT OF APPLIED MATHEMATICS & TECHNICAL MEDICINE CENTER

ISMRM ANNUAL MEETING
MAY 16, 2021

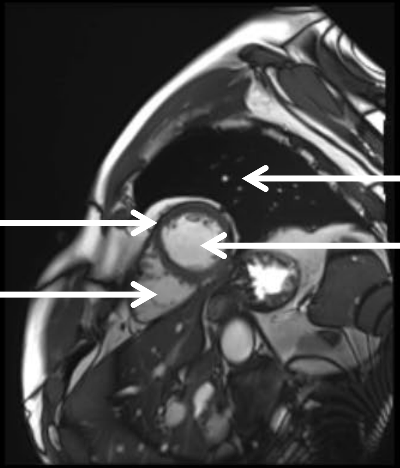
Declaration of Financial Interests or Relationships

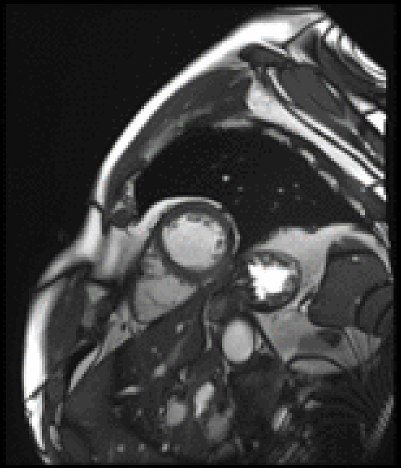
Jelmer Wolterink:

I have no financial interests or relationships to disclose with regard to the subject matter of this presentation.

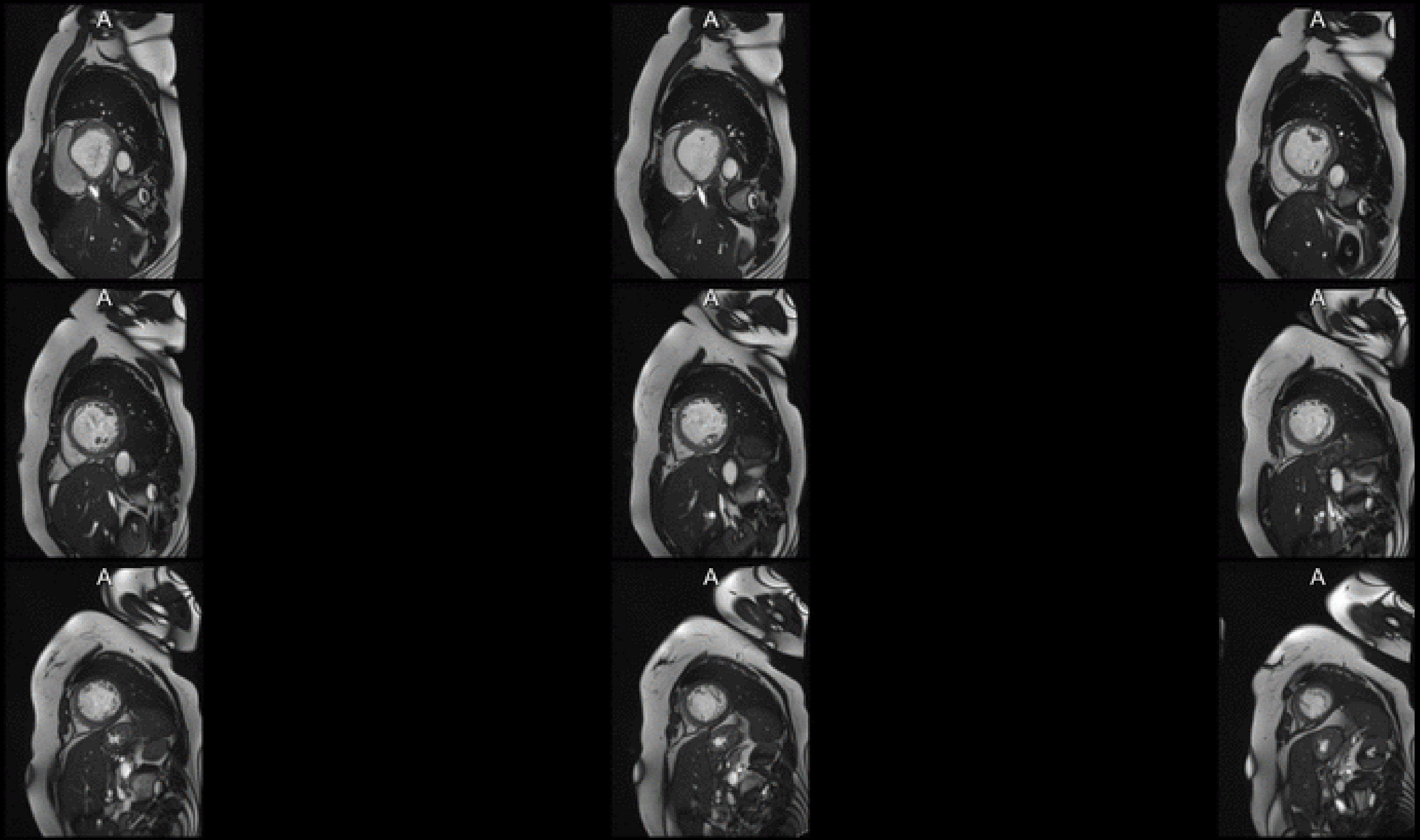
CONTENT

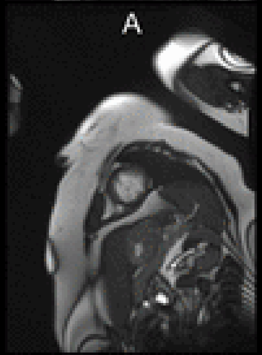
- Hands-on tutorial cardiac cine MR segmentation
- In this presentation
    1. Data and task
    2. Crash course deep learning
    3. Getting started with Python + PyTorch + MONAI


Lung
Left ventricle myocardium
Left ventricle cavity
Right ventricle



Ejection fraction: 67.2%
Normal



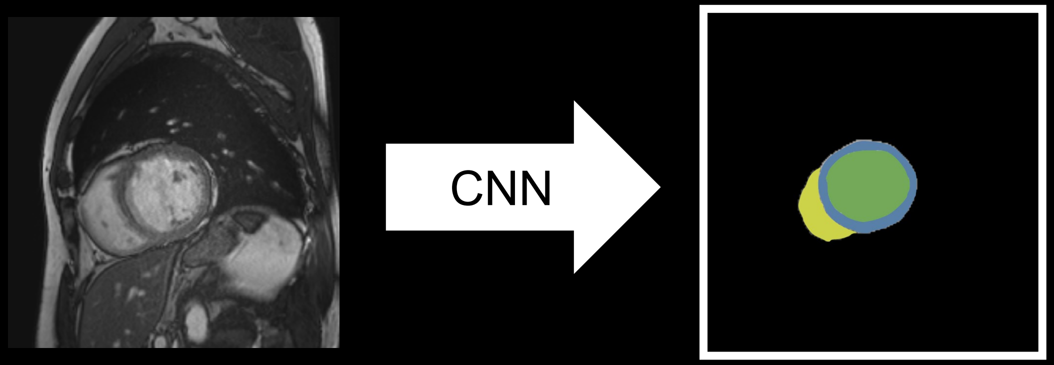


Ejection fraction: 24.4%
Heart failure

TASK IN THIS TUTORIAL

Train a convolutional neural network (CNN) for automatic segmentation in cardiac cine MRI of

- left ventricle cavity
- left ventricle myocardium
- right ventricle

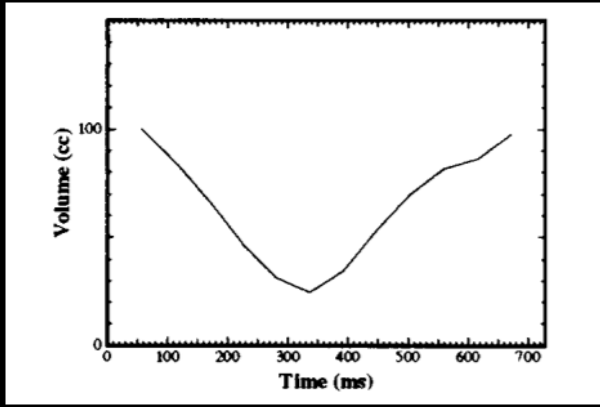


AUTOMATIC SEGMENTATION: 1993

Cardiac MR Image Segmentation Using Deformable Models

A Gupta, L von Kurowski, A Singh, D Geiger, C-C Liang, M-Y Chiu*
LP Adler, M Haacke, and DL Wilson†

* Siemens Corp. Research, Princeton, USA † Case Western Reserve Univ., Cleveland, USA



UNIVERSITY OF TWENTE.

Gupta, A. (1993). Cardiac MR image segmentation using deformable models. In *Proceedings of Computers in Cardiology Conference* (pp. 747-750). IEEE.

REVIEW: 2011

Medical Image Analysis 15 (2011) 169–184

Contents lists available at ScienceDirect

Medical Image Analysis

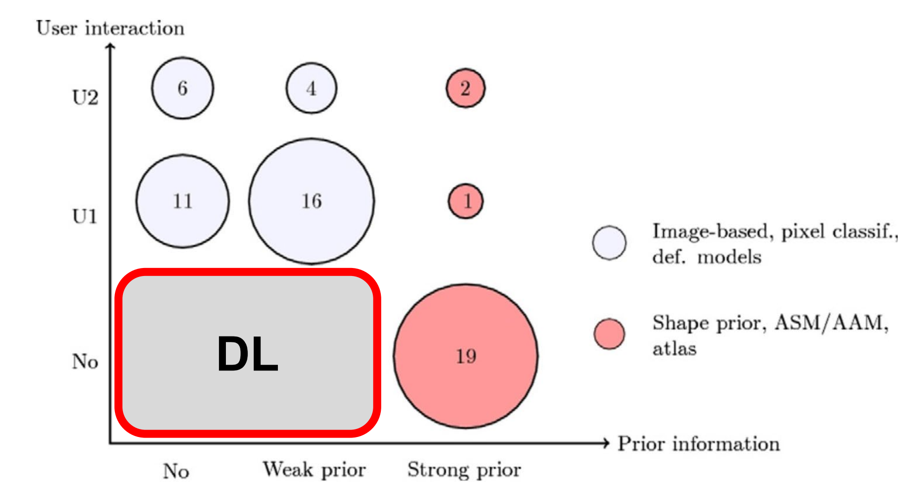
journal homepage: www.elsevier.com/locate/media

Survey Paper

A review of segmentation methods in short axis cardiac MR images

Caroline Petitjean [a,*], Jean-Nicolas Dacher [b]

[a] Université de Rouen, LITIS EA 4108, BP 12, 76801 Saint-Etienne-du-Rouvray, France
[b] University Hospital of Rouen, Department of Radiology & University of Rouen, INSERM U644, 76031 Rouen, France



UNIVERSITY OF TWENTE.

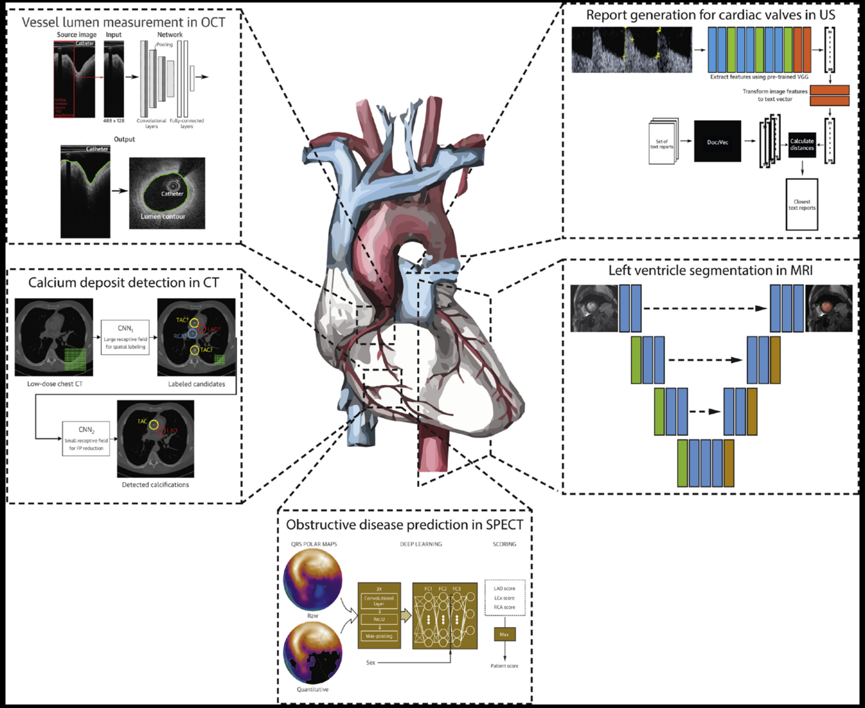
Petitjean, C., & Dacher, J. N. (2011). A review of segmentation methods in short axis cardiac MR images. *Med Imag Anal*, *15*(2), 169-184.

DEEP LEARNING FOR CVD

STATE-OF-THE-ART REVIEW

State-of-the-Art Deep Learning in Cardiovascular Image Analysis

Geert Litjens, PhD,[a] Francesco Ciompi, PhD,[a] Jelmer M. Wolterink, PhD,[b] Bob D. de Vos, PhD,[b] Tim Leiner, MD, PhD,[d] Jonas Teuwen, PhD,[c,e] Ivana Išgum, PhD[b]



UNIVERSITY OF TWENTE.

Litjens, G., Ciompi, F., Wolterink, J. M., de Vos, B. D., Leiner, T., Teuwen, J., & Išgum, I. (2019). State-of-the-art deep learning in cardiovascular image analysis. JACC: CI, 12(8 Part 1), 1549-1565.

PROBLEM SOLVED? 2018

2514 IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. 37, NO. 11, NOVEMBER 2018

Deep Learning Techniques for Automatic MRI Cardiac Multi-Structures Segmentation and Diagnosis: Is the Problem Solved?

Olivier Bernard, Alain Lalande, Clement Zotti, Frederick Cervenansky, Xin Yang, Pheng-Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel Angel Gonzalez Ballester, Gerard Sanroma, Sandy Napel, Steffen Petersen, Georgios Tziritas, Elias Grinias, Mahendra Khened, Varghese Alex Kollerathu, Ganapathy Krishnamurthi, Marc-Michel Rohé, Xavier Pennec, Maxime Sermesant, Fabian Isensee, Paul Jäger, Klaus H. Maier-Hein, Peter M. Full, Ivo Wolf, Sandy Engelhardt, Christian F. Baumgartner, Lisa M. Koch, Jelmer M. Wolterink, Ivana Išgum, Yeonggul Jang, Yoonmi Hong, Jay Patravali, Shubham Jain, Olivier Humbert, and Pierre-Marc Jodoin

UNIVERSITY OF TWENTE.

Bernard, O. et al.(2018). Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: is the problem solved?. *IEEE TMI*, *37*(11), 2514-2525.

PARTIALLY

Methods *	ED						ES					
	LV		*RV*		*MYO*		*LV*		*RV*		*MYO*	
	D	d_H	D	d_H	D	d_H	D	d_H	D	d_H	D	d_H
	val.	mm	val.	mm	val.	mm	val.	mm	val.	mm	val.	mm
O_{1a} vs O_2 (inter-obs)	0.956	5.6	0.930	12.6	0.870	6.7	0.898	8.1	0.866	14.0	0.891	7.6
O_2 vs O_{1b} (inter-obs)	0.950	6.2	0.931	12.1	0.868	7.2	0.895	8.5	0.861	14.1	0.886	8.0
O_{1a} vs O_{1b} (intra-obs)	0.967	4.0	0.957	7.6	0.900	5.1	0.941	5.4	0.930	9.1	0.917	6.0
Average DL methods vs GT	**0.965**	7.6	**0.947**	13.2	**0.906**	10.1	**0.927**	9.2	**0.886**	15.2	**0.898**	10.9
Isensee *et al.* [44] vs GT	**0.968**	7.4	**0.946**	**10.1**	**0.902**	8.7	**0.931**	**6.9**	**0.906**	**12.1**	**0.919**	8.7
O_{1a} vs O_2 (inter-obs)	0.956	4.4	0.938	7.7	0.867	5.0	0.913	5.5	0.890	8.7	0.894	5.5
O_2 vs O_{1b} (inter-obs)	0.953	4.9	0.937	8.6	0.864	5.5	0.905	5.8	0.898	9.4	0.886	6.1
O_{1a} vs O_{1b} (intra-obs)	0.971	3.1	0.960	5.8	0.905	3.6	0.950	3.9	0.940	6.9	0.923	4.4
Average DL methods vs GT	**0.972**	**3.7**	**0.951**	**8.1**	**0.896**	**5.2**	**0.929**	**4.2**	**0.899**	9.9	**0.915**	**6.1**
Isensee *et al.* [44] vs GT	**0.972**	**3.7**	**0.969**	**6.4**	**0.910**	**4.6**	**0.945**	**4.2**	**0.912**	**8.6**	**0.930**	**5.1**

* *ED: End diastole; ES: End systole; LV: Endocardial contour of the left ventricle; RV: Endocardial contour of*
* *the right Ventricle; Myo: Myocardium contours; D: Dice Index; d_H: Hausdorff distance; GT: Ground-truth.*

Bernard, O. et al.(2018). Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: is the problem solved?. *IEEE TMI*, *37*(11), 2514-2525.

NOT ENTIRELY

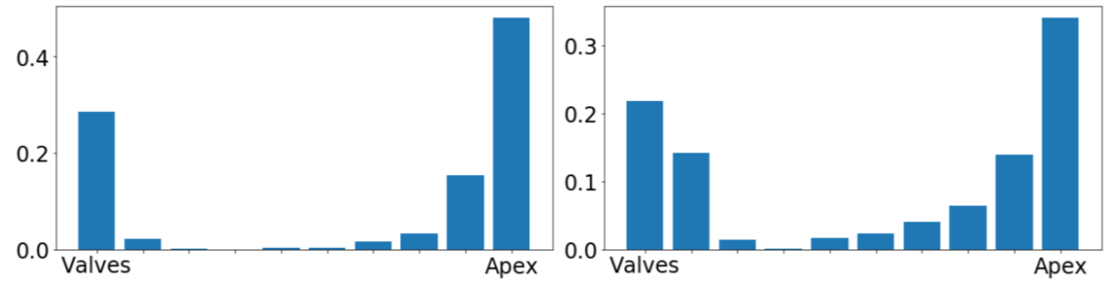


Fig. 3. Histogram of degenerated slices ED (left), and ES (right).

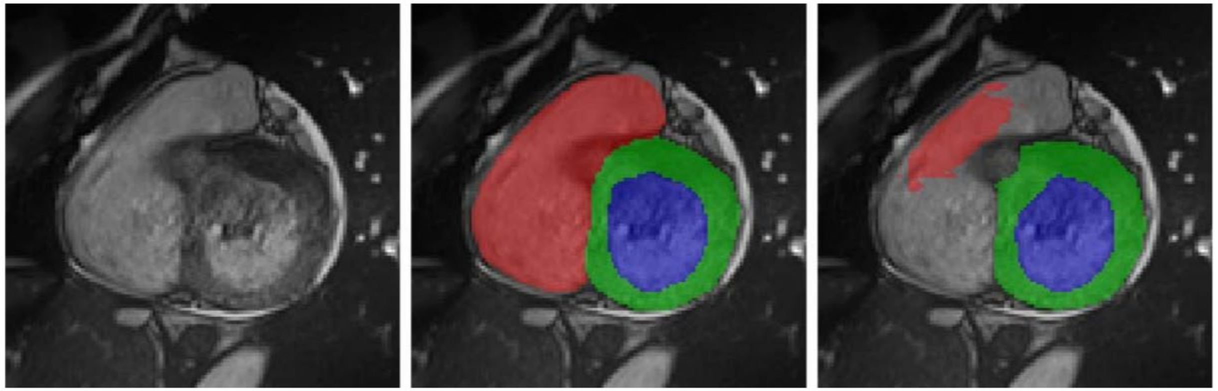


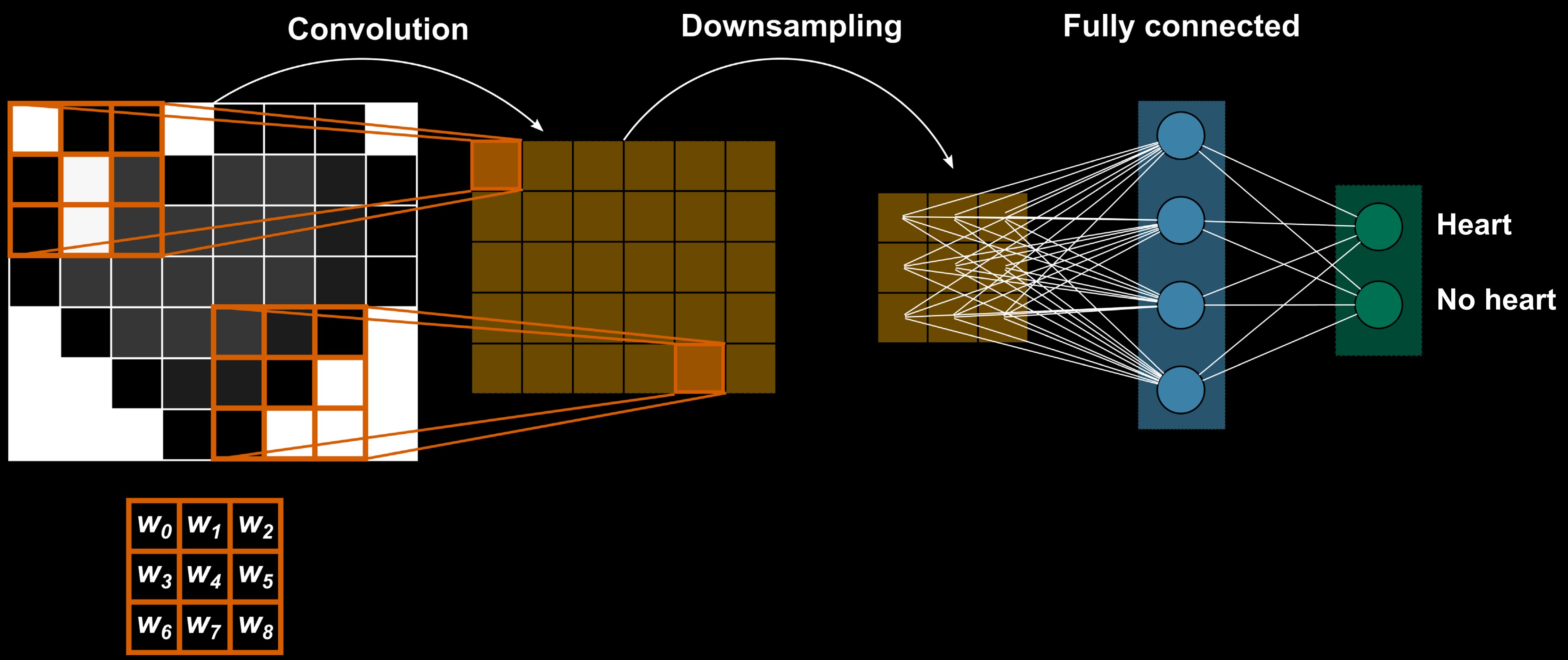
Fig. 4. Typical degenerated result at the base of the heart. [Left] input image; [Middle] ground truth; [Right] prediction.

Bernard, O. et al.(2018). Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: is the problem solved?. *IEEE TMI*, *37*(11), 2514-2525.

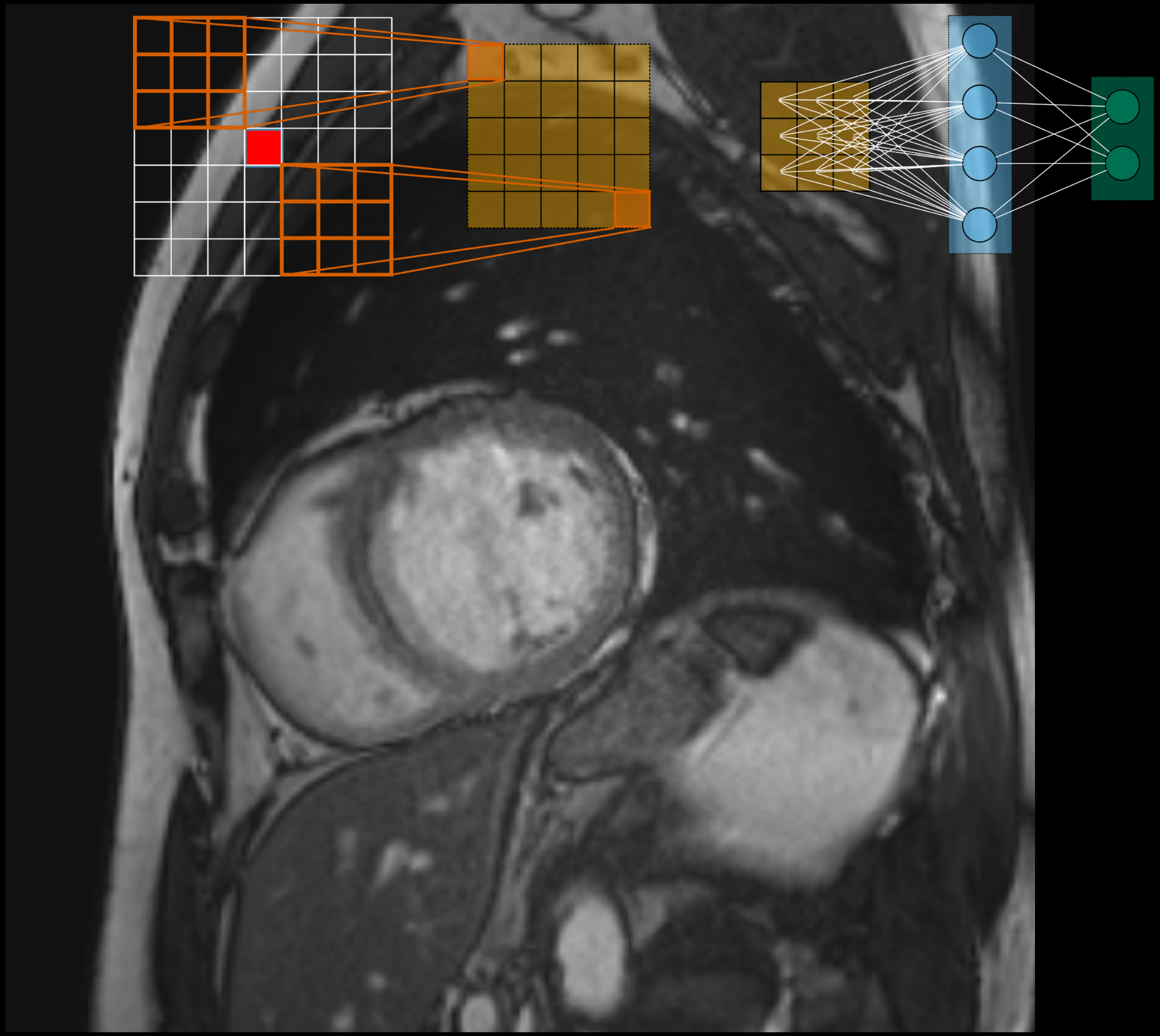
NEURAL NETWORK TRAINING

1. Training stage
   - **Optimize** the parameters of the **network**
   - Minimize **loss**, backpropagation of errors
   - Required: **images** with reference **segmentations** (training set)
2. Test stage
   - Apply the trained networks
   - Required: images with reference segmentations (test set)

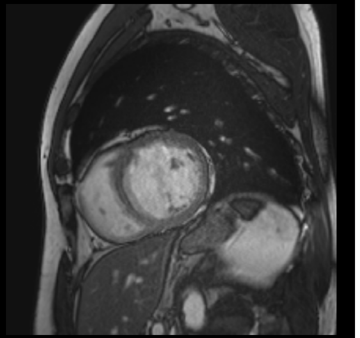
CONVOLUTIONAL NEURAL NETWORK (CNN)

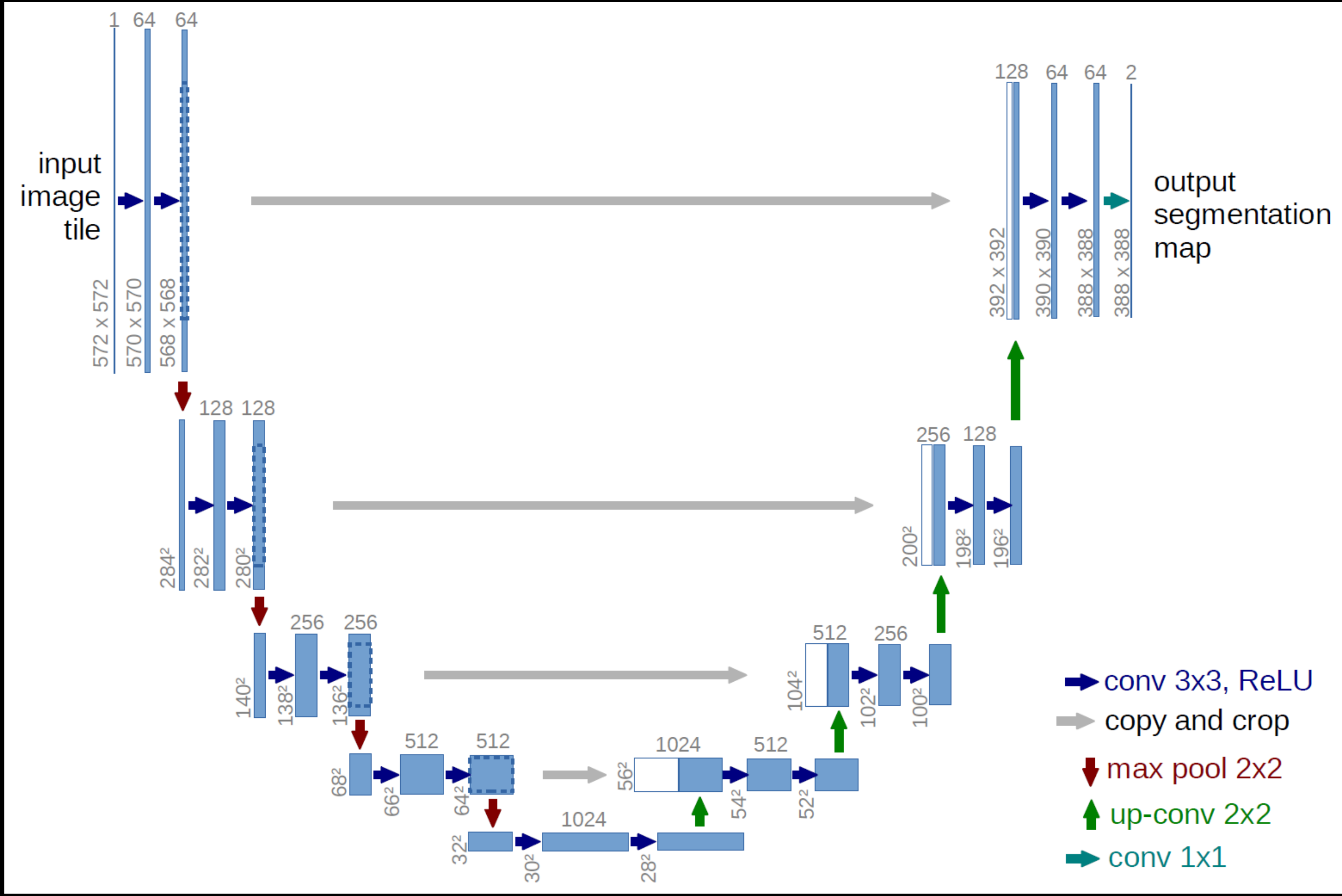


SLIDING-WINDOW PIXEL CLASSIFICATION



2D U-NET

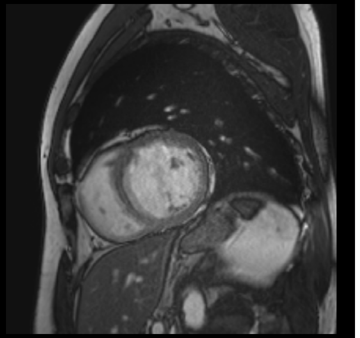


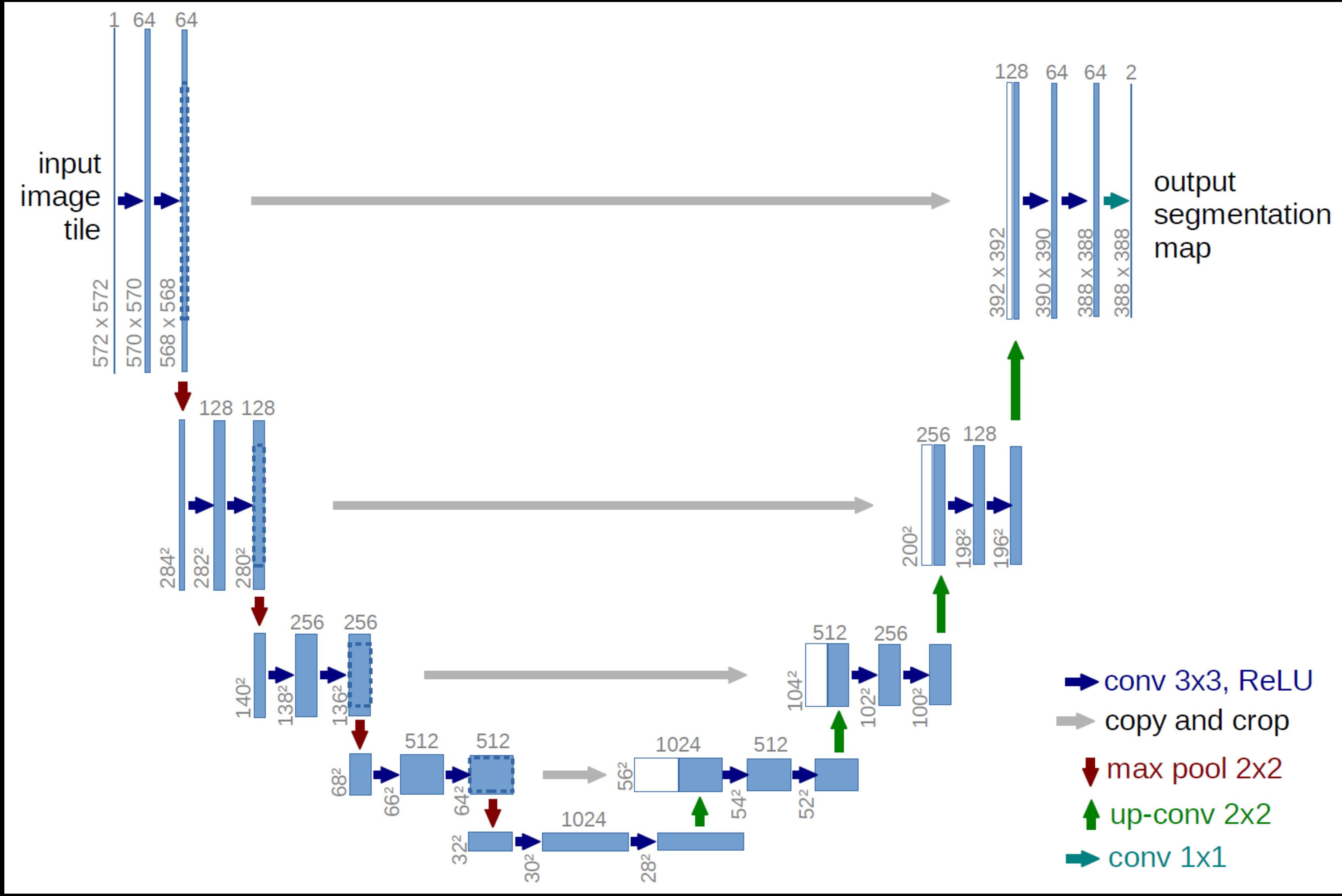


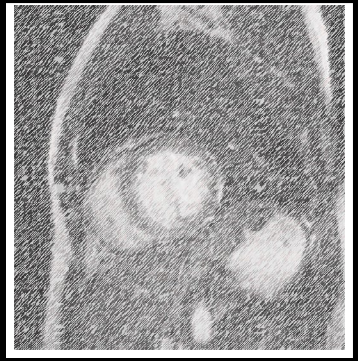
UNIVERSITY
OF TWENTE.

Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *MICCAI* (pp. 234-241). Springer, Cham.

2D U-NET







UNIVERSITY OF TWENTE.

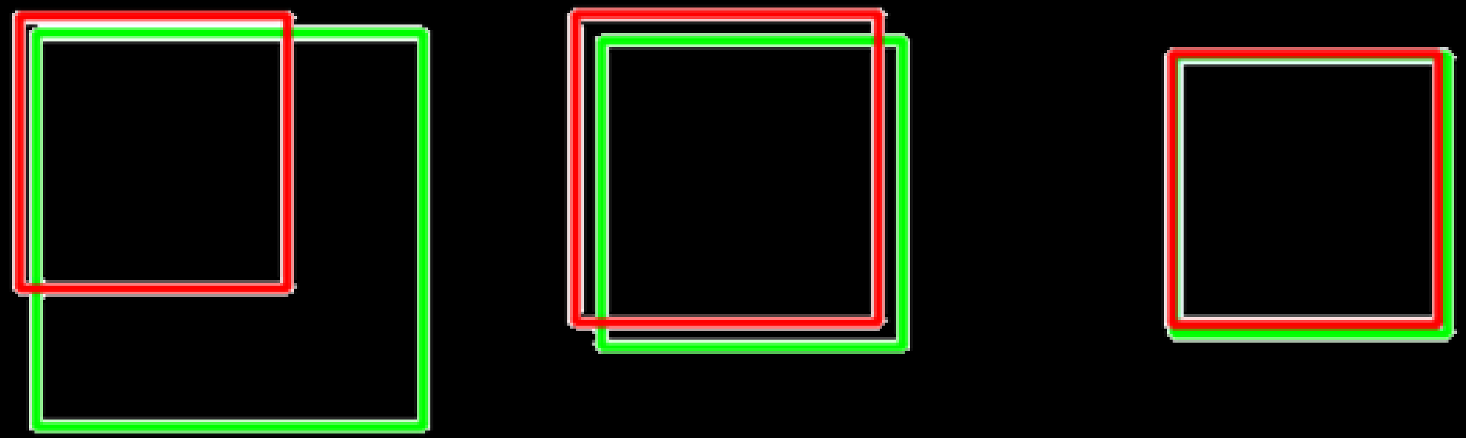
Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *MICCAI* (pp. 234-241). Springer, Cham.

DICE LOSS

Based on Dice similarity coefficient (DSC)

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

DSC = 0.57 DSC = 0.83 DSC = 0.96



Poor Good Excellent

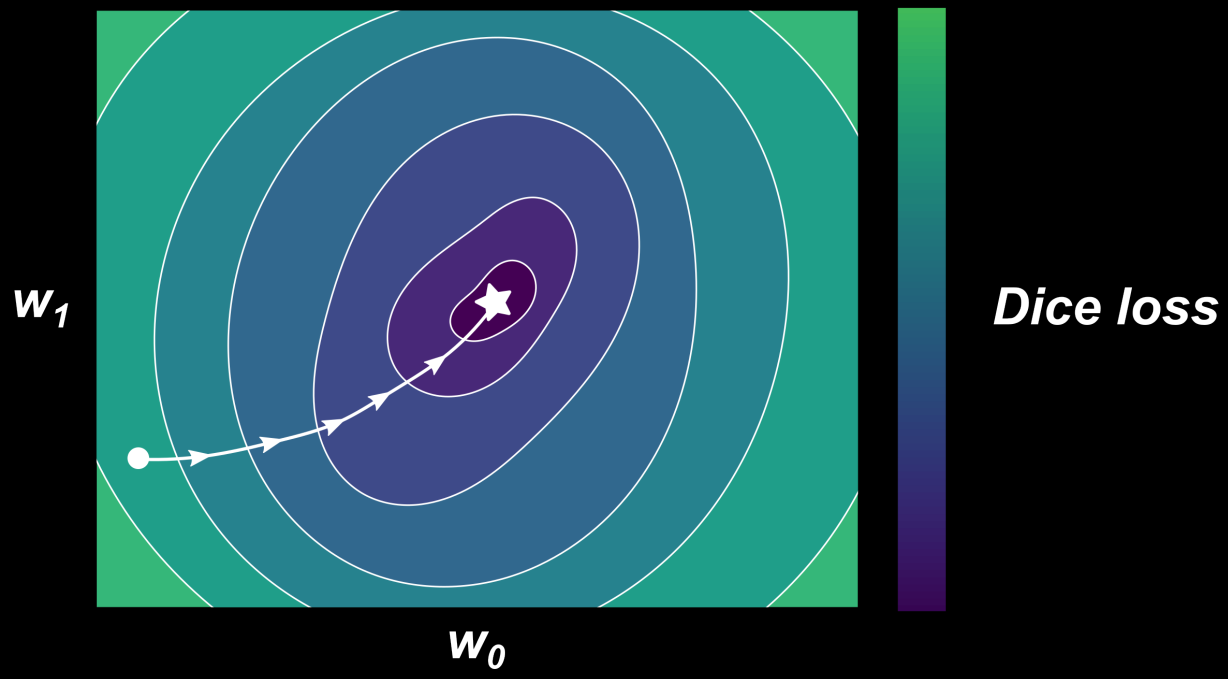
UNIVERSITY OF TWENTE.

Milletari, F., Navab, N., & Ahmadi, S. A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV* (pp. 565-571).

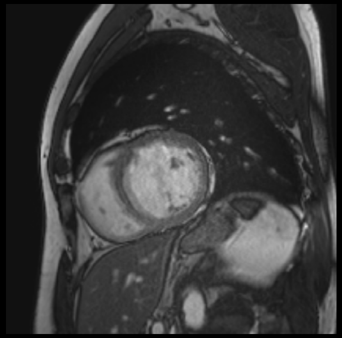
OPTIMIZATION

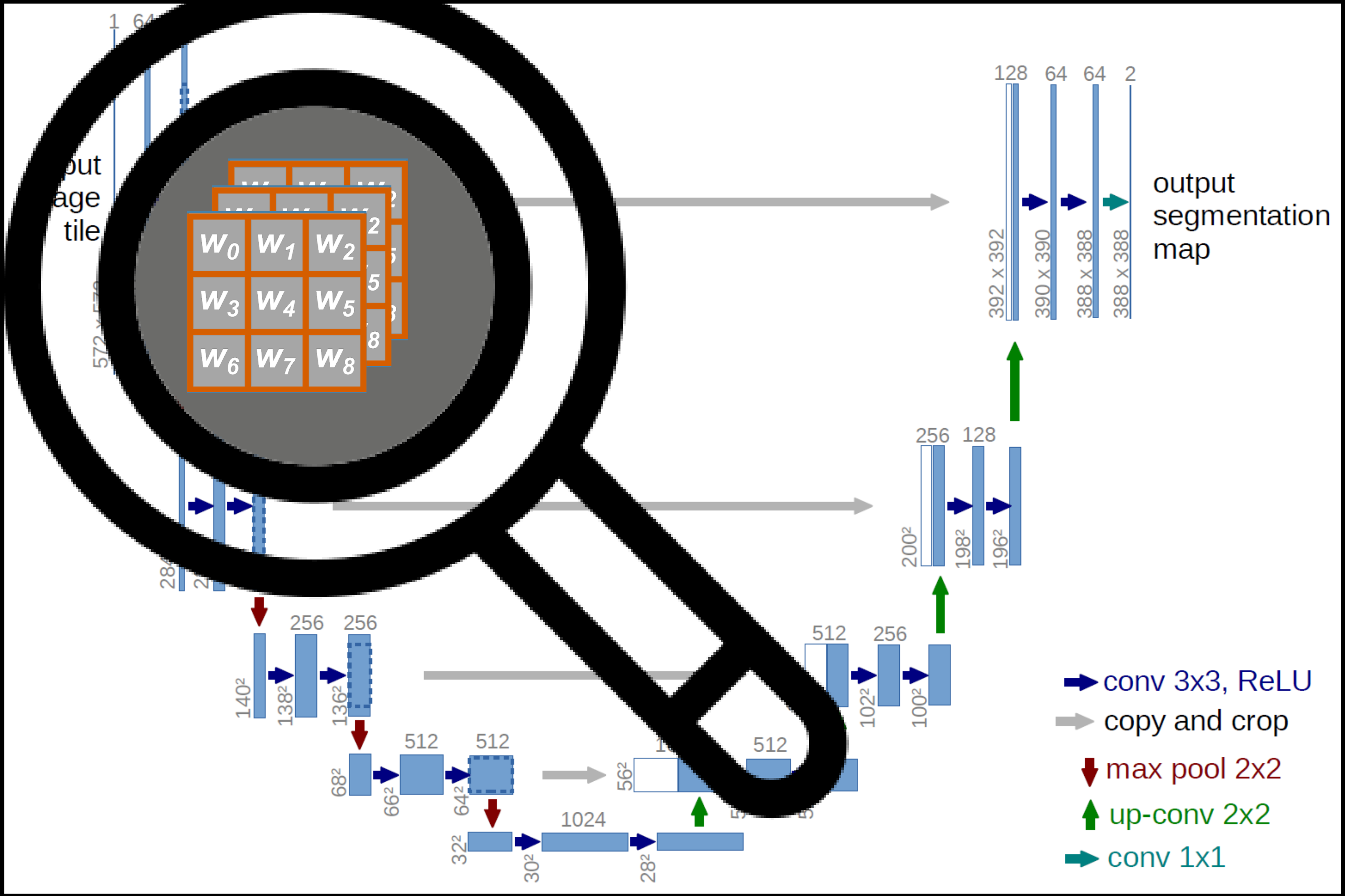
Iterative process, in each iteration

1. Take random ‘mini-batch’ of samples
2. Obtain prediction on samples and compute loss w.r.t. reference
3. Backpropagate loss
4. Update weights
5. Repeat



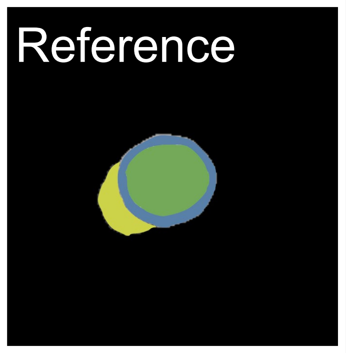
BACKPROPAGATION





Dice loss

Reference



UNIVERSITY OF TWENTE.

Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *MICCAI* (pp. 234-241). Springer, Cham.

SUMMARY

- What do we (at least) need
    1. A data set
    2. A neural network architecture
    3. A loss function
    4. An optimization algorithm

DATA SET

- Sunnybrook cardiac MR data set[1]
  - 45 patients, 4 pathologies
- Multi-Centre, Multi-Vendor & Multi-Disease Cardiac Image Segmentation Challenge (M&Ms)[2]
  - 350 patients, 4 scanners, 8 pathologies
- Automatic Cardiac Diagnosis Challenge (ACDC)[3]
  - 150 patients, 4 pathologies

[1] https://www.cardiacatlas.org/studies/sunnybrook-cardiac-data/
[2] https://www.ub.edu/mnms/
[3] https://www.creatis.insa-lyon.fr/Challenge/acdc/

DATA SET

- Sunnybrook cardiac MR data set[1]
  - 45 patients, 4 pathologies
- Multi-Centre, Multi-Vendor & Multi-Disease Cardiac Image Segmentation Challenge (M&Ms)[2]
  - 350 patients, 4 scanners, 8 pathologies
- Automatic Cardiac Diagnosis Challenge (ACDC)[3]
  - 150 patients, 4 pathologies

[1] https://www.cardiacatlas.org/studies/sunnybrook-cardiac-data/
[2] https://www.ub.edu/mnms/
[3] https://www.creatis.insa-lyon.fr/Challenge/acdc/

ACDC DATA SET

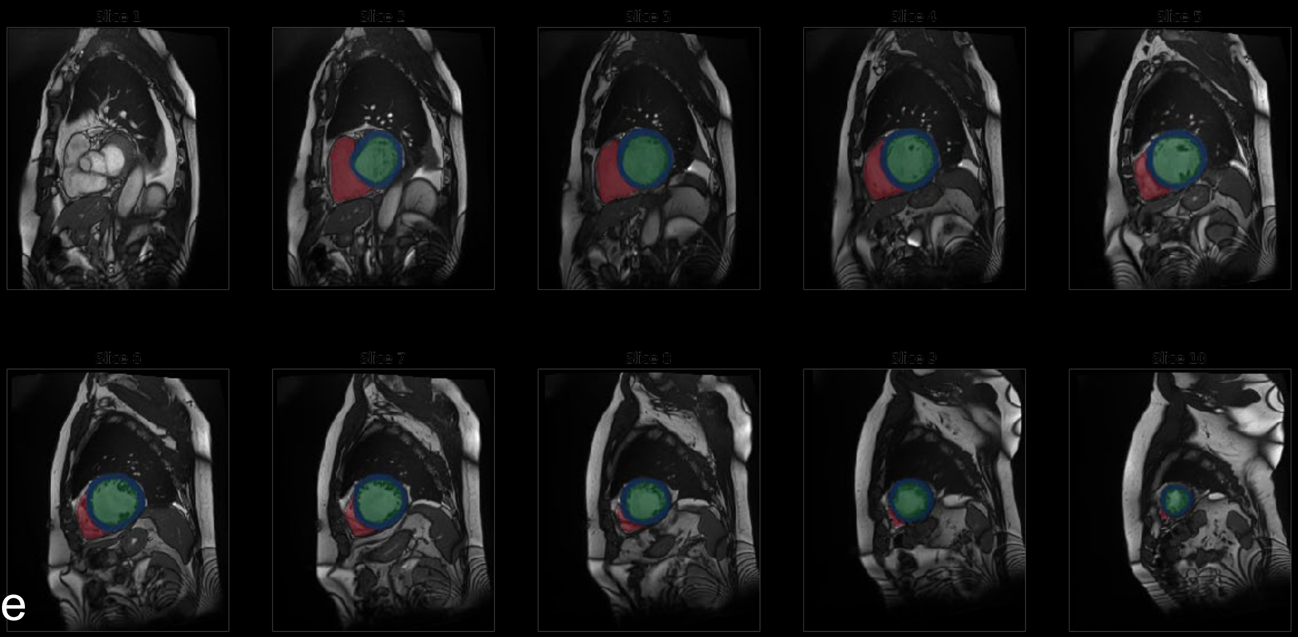
For each patient

- 20-30 frames
- Annotations in end-diastole/end-systole frames

Training set: 100 patients

Test set: 50 patients

Pt 1, end-diastole



Left ventricle

Myocardium

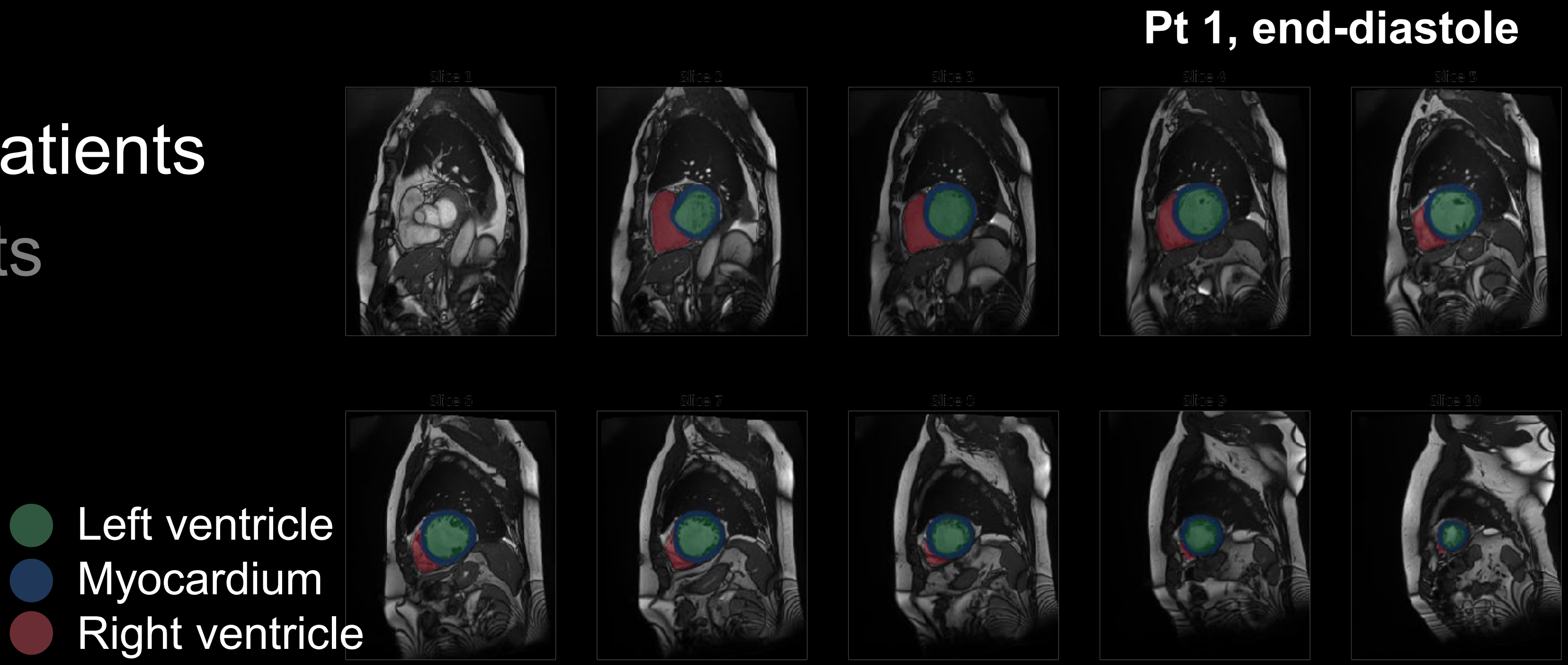
Right ventricle

ACDC DATA SET

For each patient
- 20-30 frames
- Annotations in end-diastole/end-systole frames

Training set: 100 patients

Test set: 50 patients



PYTHON GETTING STARTED

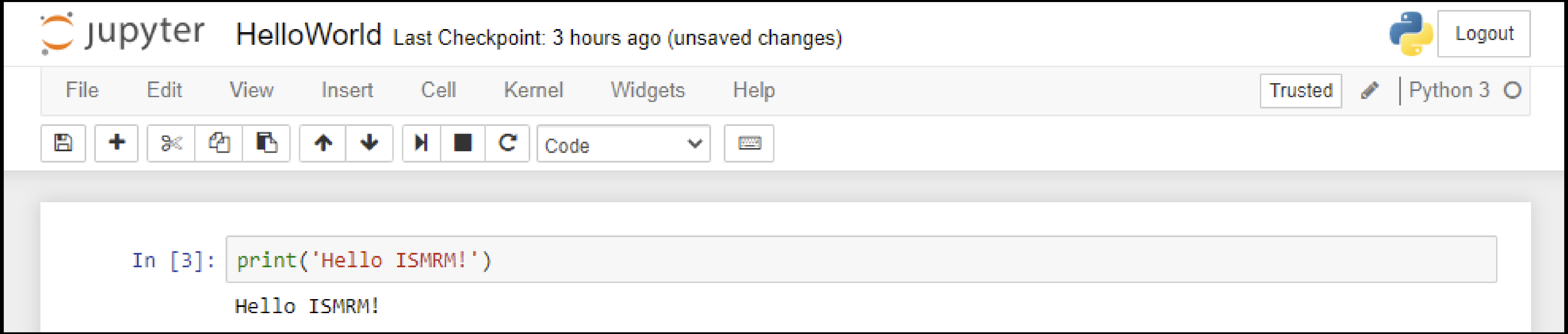
1. Install a Python distribution: Anaconda[1]
2. Install a Python editor: JupyterLab[2]
3. Start Jupyter

```
$ jupyter notebook
```

[1] https://www.anaconda.com/products/individual
[2] https://jupyter.org/, https://jupyterlab.readthedocs.io/en/stable/

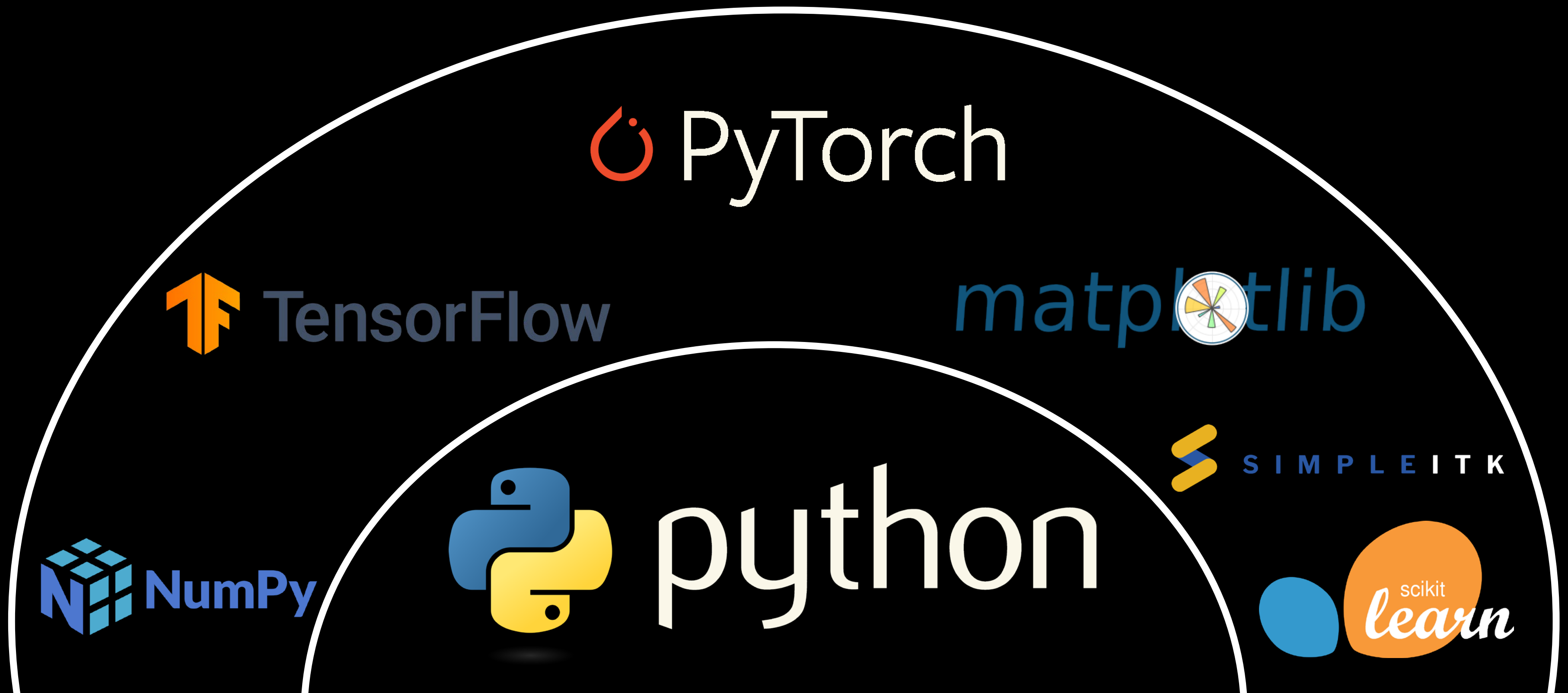
PYTHON GETTING STARTED

Jupyter allows you to edit and run Python in a browser



[1] https://www.anaconda.com/products/individual
[2] https://jupyter.org/

PYTHON PACKAGES

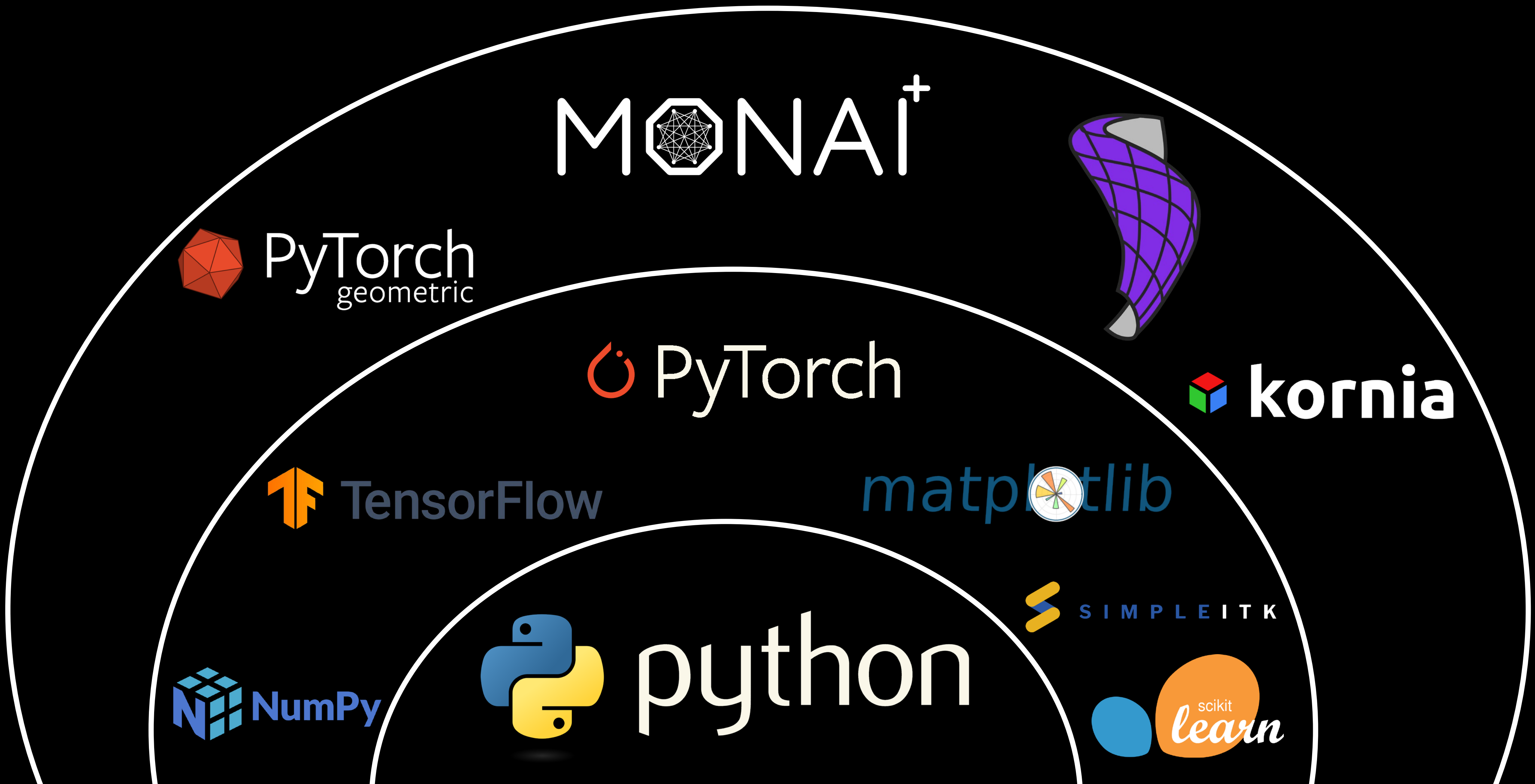


PYTORCH[1]

- Provides important deep learning building blocks
  - Optimizers
  - Neural network layers
  - Running on different devices
  - Data augmentation
  - Data loaders
  - …
- Very versatile, well-documented[2]

[1] https://pytorch.org/
[2] https://pytorch.org/tutorials/beginner/basics/intro.html

PYTORCH ECOSYSTEM



MONAI[1]

- PyTorch for healthcare data
- Medical image loaders
  - NIfTI, DICOM, MHD, …
- Medical imaging networks
  - U-Net, V-Net, …
- Medical imaging data augmentation
  - Random deformations
  - Random bias fields (for MRI)

[1] https://monai.io/

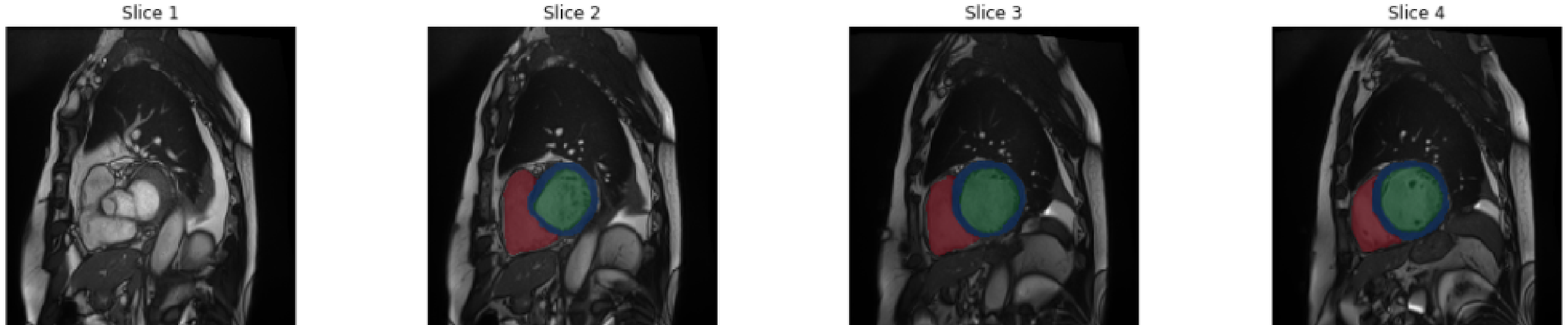
UNIVERSITY OF TWENTE.

LOADING AND VIEWING DATA

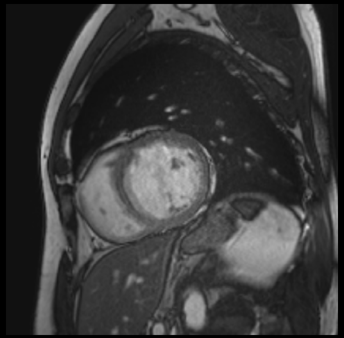
In [3]:
```
transform = monai.transforms.LoadImageD(("image", "label"))

file_dict = {"image": "{}/patient001/patient001_frame01.nii.gz".format(datapath),
             "label": "{}/patient001/patient001_frame01_gt.nii.gz".format(datapath)}

data_dict = transform(file_dict)
```

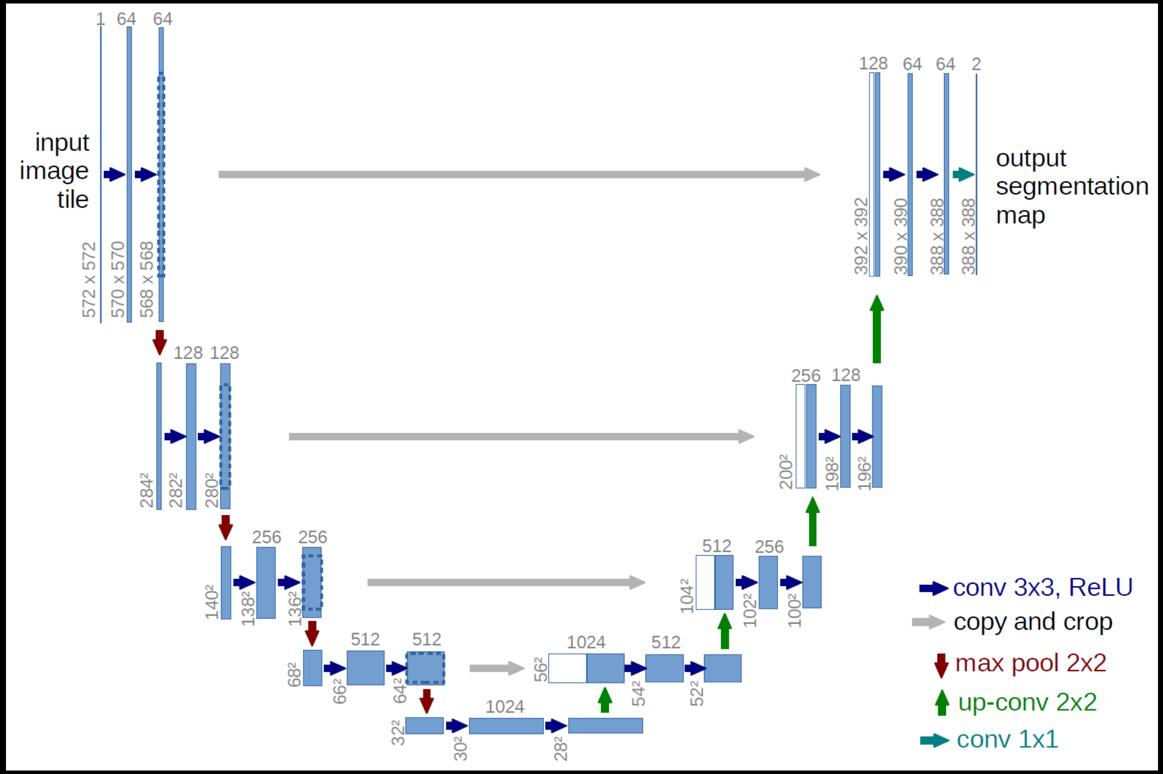
We then use Matplotlib to visualize the images in a rectangular grid. The for-loop iterates over z, the slice index. Each slice is shown, and the reference label is shown as colored overlay.

In [4]:
```
def visualize_data(pt_dict):
    plt.figure(figsize=(20,20))

    for z in range(pt_dict["image"].shape[0]):
      plt.subplot(np.ceil(np.sqrt(pt_dict["image"].shape[0])), np.ceil(np.sqrt(pt_dict["image"].shape[0])), 1 + z)
      plt.imshow(pt_dict["image"][z, :, :], cmap='gray')
      plt.axis('off')
      plt.imshow(np.ma.masked_where(pt_dict["label"][z, :, :]!=2, pt_dict["label"][z, :, :]==2), alpha=0.6, cmap='Blues', clim=(0
      plt.imshow(np.ma.masked_where(pt_dict["label"][z, :, :]!=3, pt_dict["label"][z, :, :]==3), alpha=0.6, cmap='Greens', clim=(
      plt.imshow(np.ma.masked_where(pt_dict["label"][z, :, :]!=1, pt_dict["label"][z, :, :]==1), alpha=0.6, cmap='Reds', clim=(0,
      plt.title('Slice {}'.format(z + 1))
    plt.show()

visualize_data(data_dict)
```



NETWORK





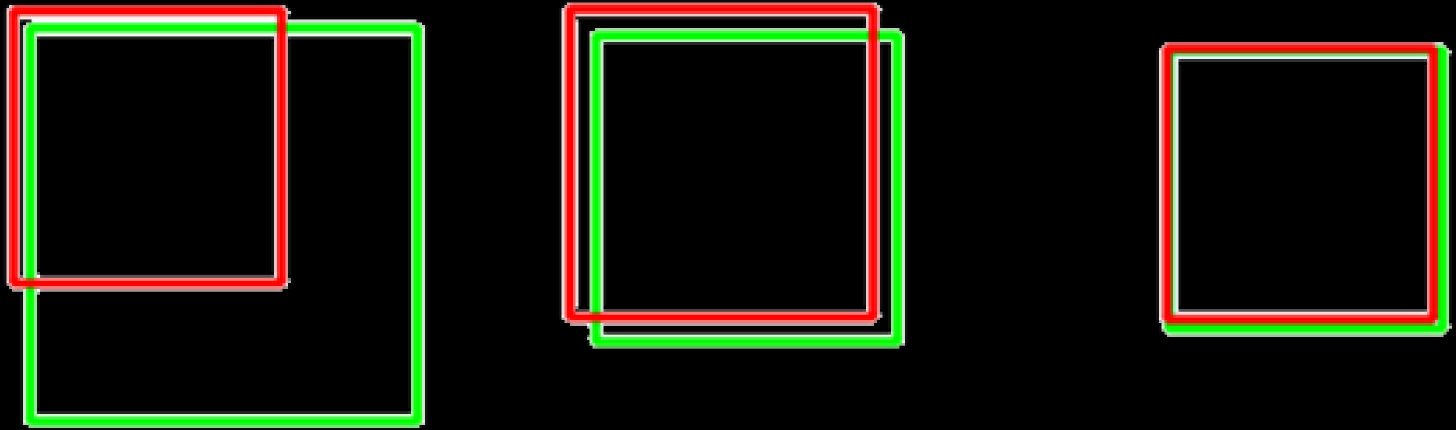
CODE

```
model = monai.networks.nets.UNet(dimensions=2,
        in_channels=1, out_channels=4,
         channels=(64, 128, 256, 512, 1024))
```

UNIVERSITY
OF TWENTE.

DICE LOSS

DSC = 0.57 DSC = 0.83 DSC = 0.96

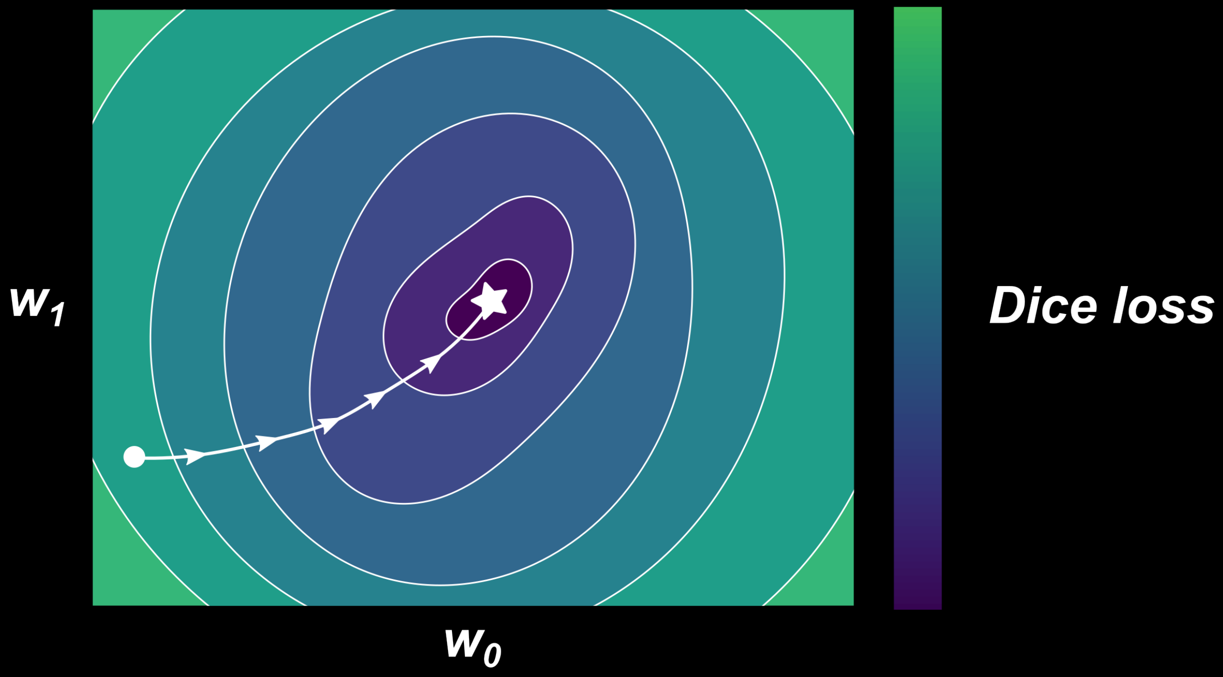


Poor Good Excellent

CODE

```
loss_function = monai.losses.DiceLoss(softmax=True)
loss = loss_function(input, target)
```

OPTIMIZER



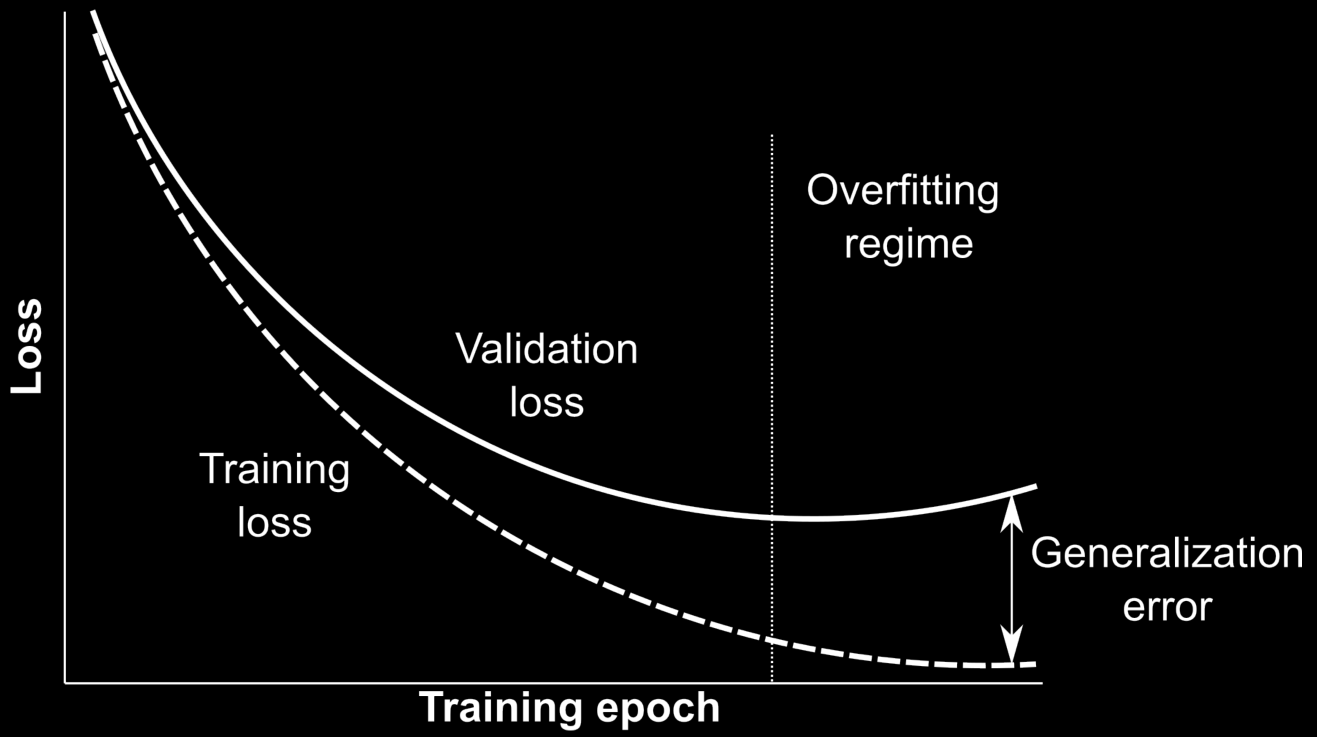
CODE

```
optimizer = torch.optim.Adam(model.parameters(), 1e-3)
loss.backward()
optimizer.step()
```

UNIVERSITY OF TWENTE.

MONITOR YOUR TRAINING PROCESS

- Monitor loss curves and identify if the model has converged
- Spot overfitting/generalization error

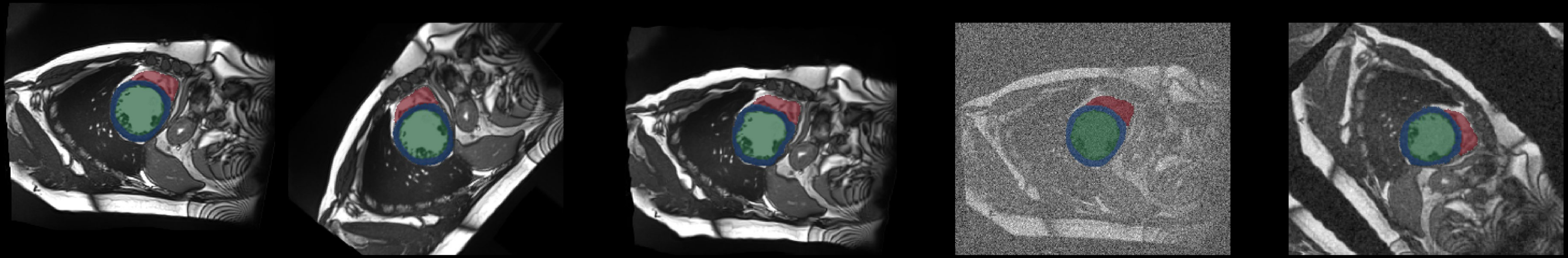


UNIVERSITY OF TWENTE.

DATA AUGMENTATION

- Add artificial samples to the training data set
- Wide range of randomized image transforms possible

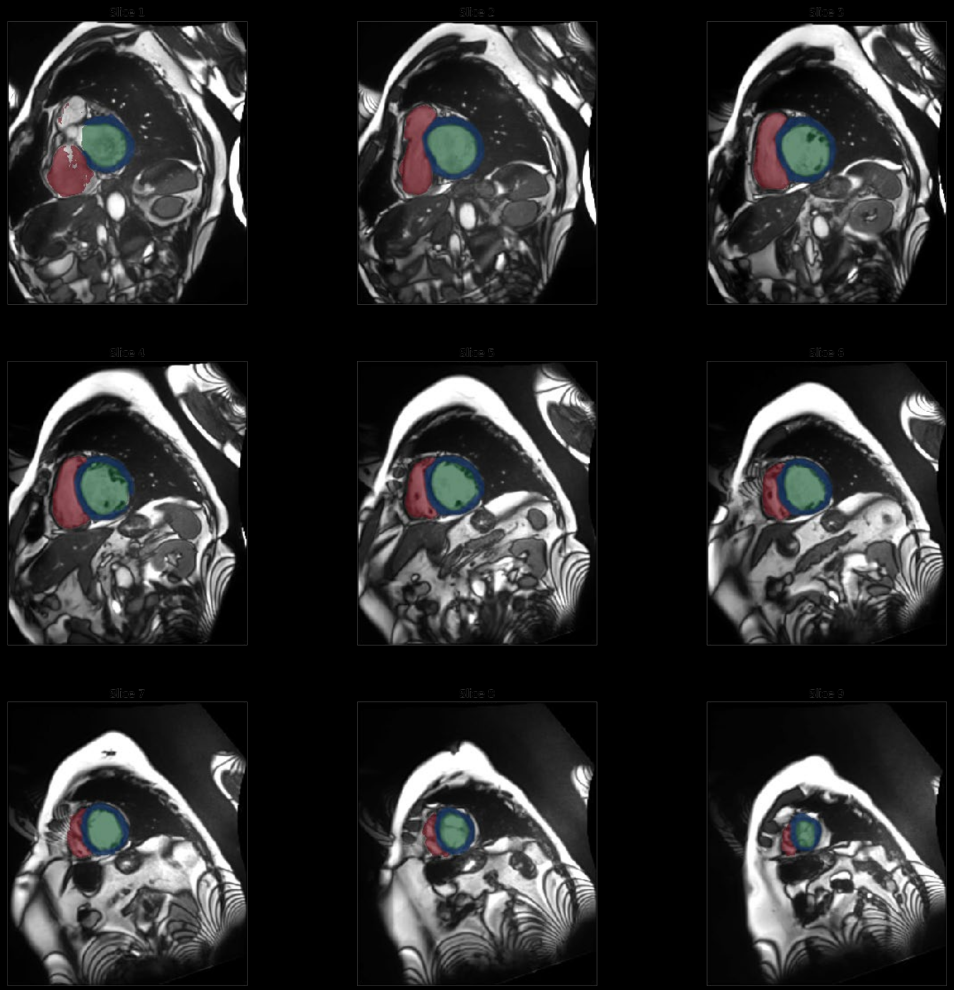
Original | Rotate | Elastic deformation | Gaussian noise | All three



VALIDATING YOUR RESULTS

Compare automatic and reference segmentations

- Dice similarity coefficient
  `monai.metrics.DiceMetric()`
- Hausdorff distance
  `monai.metrics.HausdorffDistanceMetric()`



Average DSC 0.92, average Hausdorff distance 5.97 mm

CODE

Example code on

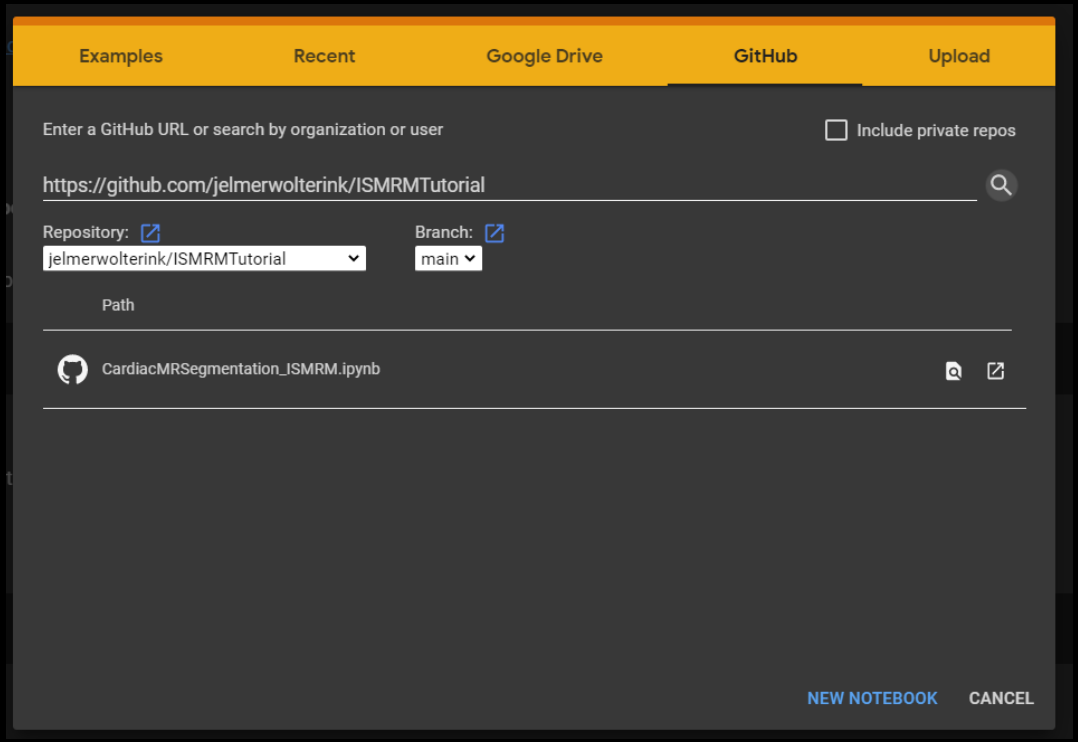
https://github.com/jelmerwolterink/ISMRMTutorial

- Loading and viewing images
- Training a U-Net in Python + PyTorch + MONAI
- Evaluating and visualizing your results

[1] https://pytorch.org/
[2] https://pytorch.org/tutorials/beginner/basics/intro.html

ALTERNATIVELY

- Google Colaboratory[1] provides Python notebooks in the cloud
- Free GPU utilization[2]
- Load notebook from GitHub



[1] https://colab.research.google.com/
[2] https://colab.research.google.com/notebooks/gpu.ipynb

WHAT CAN YOU DO?

- Training set augmentation (using PyTorch[1], MONAI[2], TorchIO[3])
- Modify U-Net architecture (3D vs. 2D)
- Other networks[4], e.g. DynUNet, VNet
- Other optimizers[5], e.g. SGD, AdaGrad, ..
- Add data from other sources, e.g. M&Ms
- Segment the full cardiac cycle in ACDC
- …

[1] https://pytorch.org/vision/stable/transforms.html
[2] https://docs.monai.io/en/latest/transforms.html
[3] https://torchio.readthedocs.io/transforms/augmentation.html
[4] https://docs.monai.io/en/latest/networks.html
[5] https://pytorch.org/docs/stable/optim.html

SUMMARY

- ACDC data set
- Python + PyTorch + MONAI
- Minimal code provided
- Q&A on Thursday, post questions as issues on GitHub

Code: https://github.com/jelmerwolterink/ISMRMTutorial

Thank you for your attention!

j.m.wolterink@utwente.nl
@jelmerwolterink

UNIVERSITY
OF TWENTE.