Post-processing smoothness prior reconstructions using CNNs

R. G. Beraldo, L. A. Ferreira, F. S. Moura, A. K Takahata, R. Suyama

Center for Engineering, Modeling and Applied Social Sciences
Federal University of ABC (Brazil)

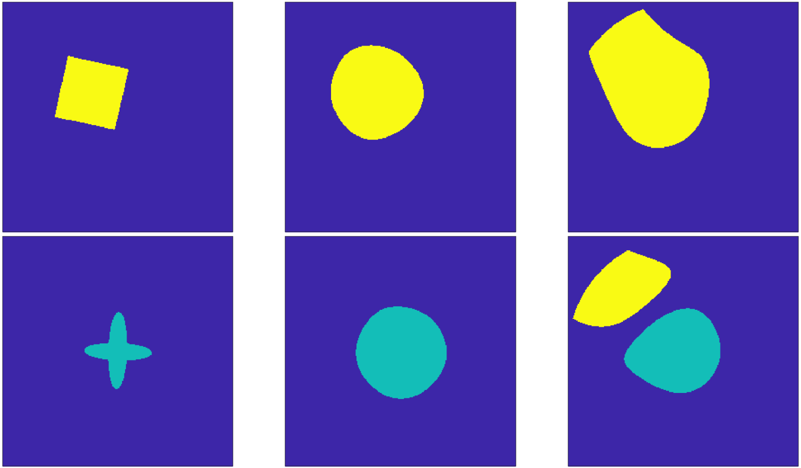
2024

Github repository

These slides are related to the code available at
https://github.com/robert-abc/KTC2023-ABC1/
to join the Kuopio Tomography Challenge 2023
https://www.fips.fi/KTC2023.php

KTC example code (Smoothness Prior) [1]

One-step reconstruction: $\Delta\sigma = (\mathbf{J}^T\Gamma_n\mathbf{J} + \mathbf{L}^T\mathbf{L})^{-1}\mathbf{J}^T\Gamma_n\Delta\mathbf{v}$



Ground truth Level 1 Level 7

We submitted two proposals:

1. Post-processing the Smoothness Prior reconstruction to extract meaningful information
    - github.com/robert-abc/KTC2023-ABC1/
    - Score: 12.75
2. Deep Image Prior with total variation regularization
    - github.com/robert-abc/KTC2023-ABC2/
    - Score: 11.25

In this slides, we illustrate the first proposal (Matlab version)

Proposal Summary (Matlab Version)

There are six steps to reconstruct and to segmentate the results:

1. Reconstruction with smoothness prior
2. Soft thresholding
3. Denoising and deblurring using a CNN
4. Thresholding
5. Opening (morphological filter)
6. Segmentation using Otsu's method

This is the original version, as submitted to KTC2023.

Step-by-step visualization

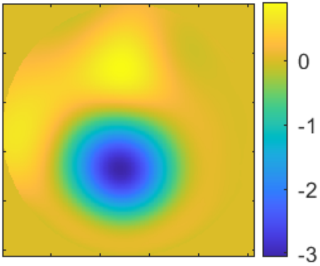
Next slides: Case 2, level 7 from the training set.

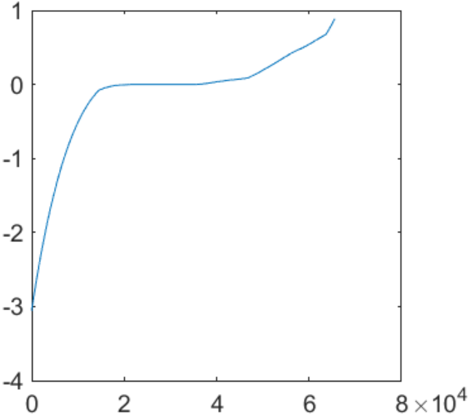
The figures on the left are the outputs of each step.

The figures on the right are the sorted visualizations, in ascending order, of the pixel values from the figures on the left.

Step 1/6: Smoothness Prior

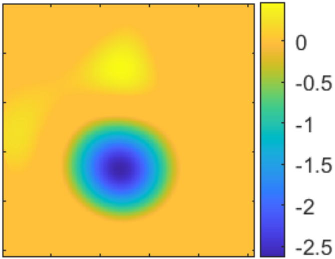
One-step reconstruction: $\Delta\sigma = (\mathbf{J}^T\Gamma_n\mathbf{J} + 20\mathbf{L}^T\mathbf{L})^{-1}\mathbf{J}^T\Gamma_n\Delta\mathbf{v}$

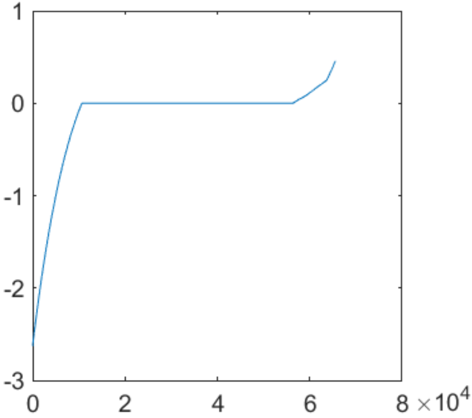




Step 2/6: Soft thresholding

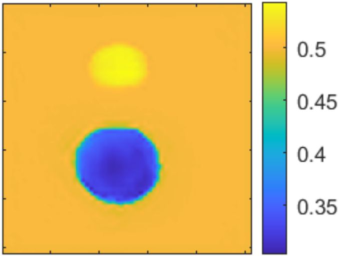
Threshold value: Proportional to the highest value of $\Delta\sigma$ from Step 1: $T = max(abs(\Delta\sigma)) * 0.14$

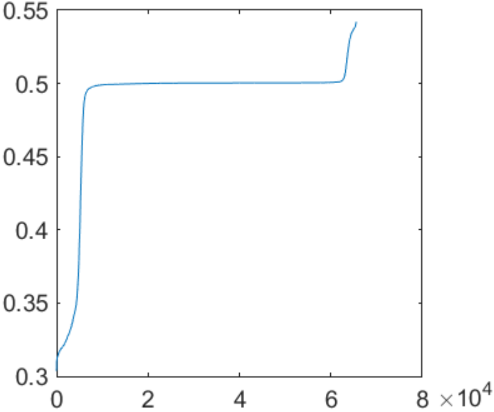




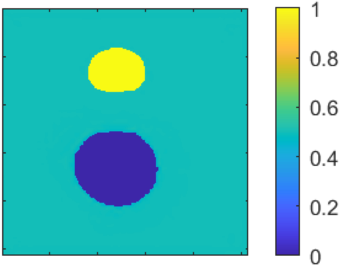
Step 3/6: Denoising and deblurring using a CNN

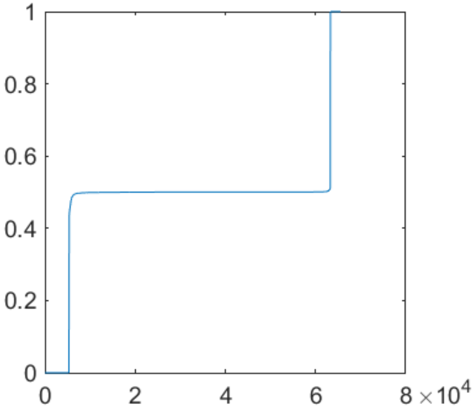
See slides 13-20 for CNN details





Step 4/6: New thresholding

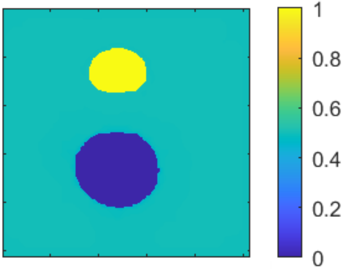


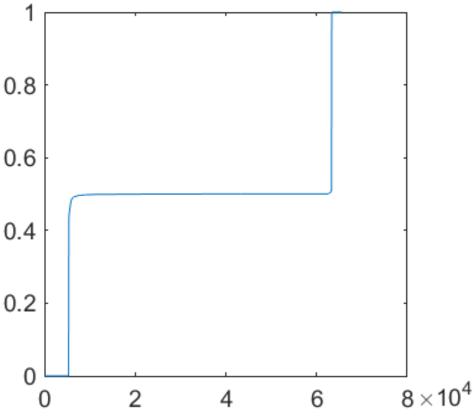


- c1 $=$ (max($\Delta\sigma$) - 0.50)./4;
- $\Delta\sigma(\Delta\sigma{>}0.5 + \text{c1}) = 1$;

- c2 $=$ (min($\Delta\sigma$) - 0.50)./4;
- $\Delta\sigma(\Delta\sigma < 0.48 + \text{c2}) = 0$;

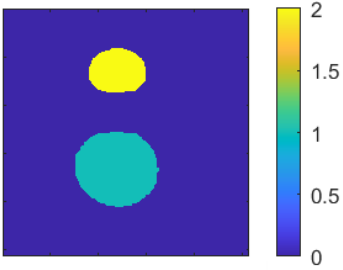
Step 5/6: Opening morphological filter

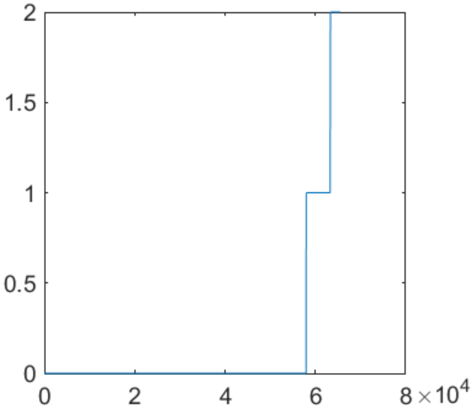




Step 6/6: Segmentation using Otsu's method

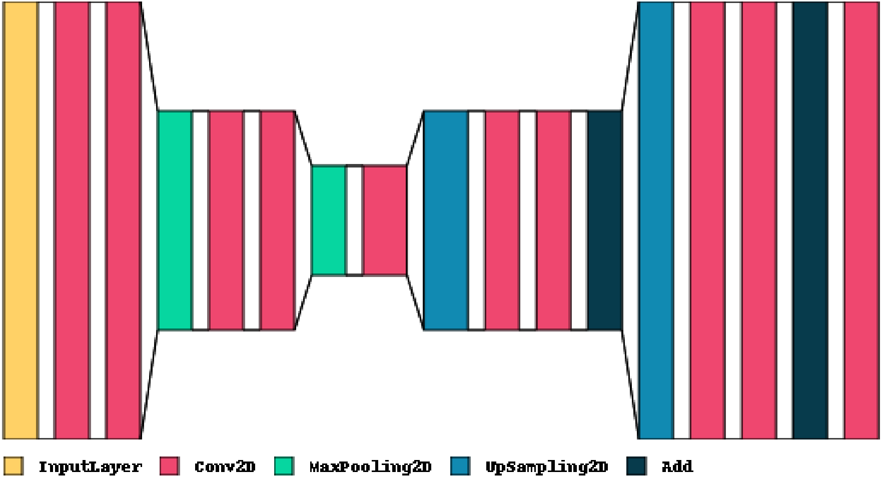
After the segmentation, the class are attributed to the targets (2 for conductive, 1 for resistive).





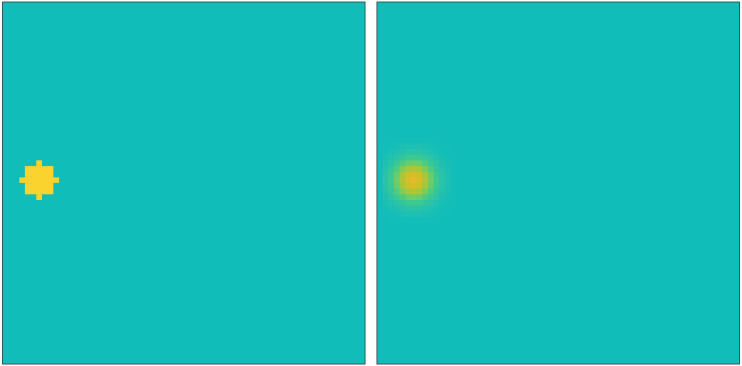
CNN details

CNN architecture (See [2] for more information)



CNN: Training pair examples (Available at [2])

Deblurring-only



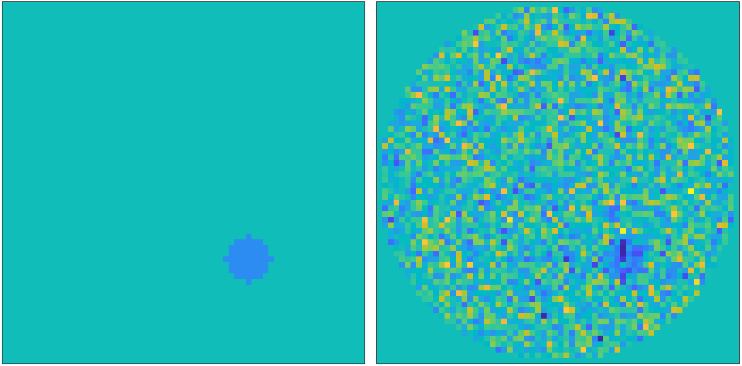
(a) $\Delta\boldsymbol{\sigma}_1$

(b) $\mathbf{h}_g * \Delta\boldsymbol{\sigma}_1$

Where $\mathbf{h}_g$ is a Gaussian kernel.

CNN: Training pair examples (Available at [2])

Denoising-only



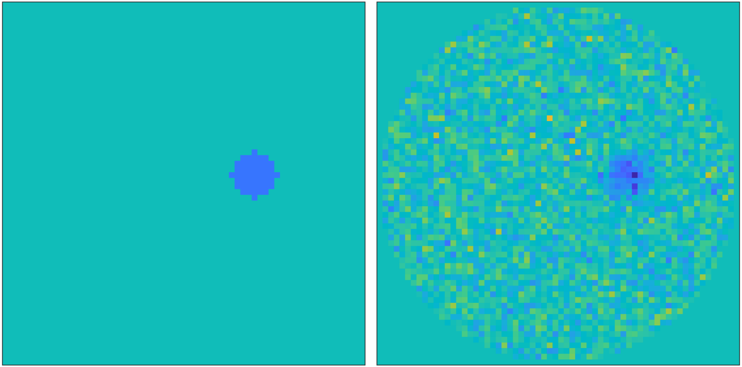
(a) $\Delta\boldsymbol{\sigma}_2$

(b) $\Delta\boldsymbol{\sigma}_2 + \boldsymbol{\delta}$

Where $\boldsymbol{\delta}$ is a Gaussian noise.

CNN: Training pair examples (Available at [2])

Denoising and deblurring

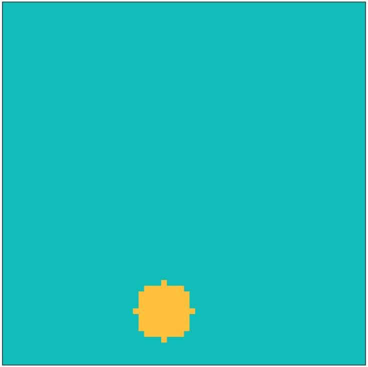


(a) $\Delta\boldsymbol{\sigma}_3$

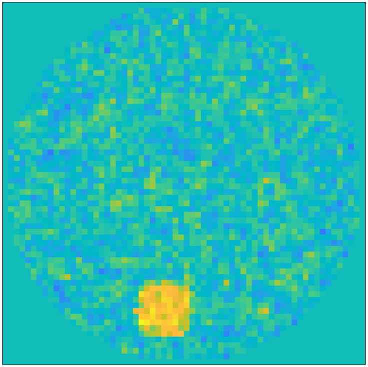
(b) $(\mathbf{h}_g * \Delta\boldsymbol{\sigma}_3) + \boldsymbol{\delta}$

CNN: Training pair examples (Available at [2])

Denoising and deblurring



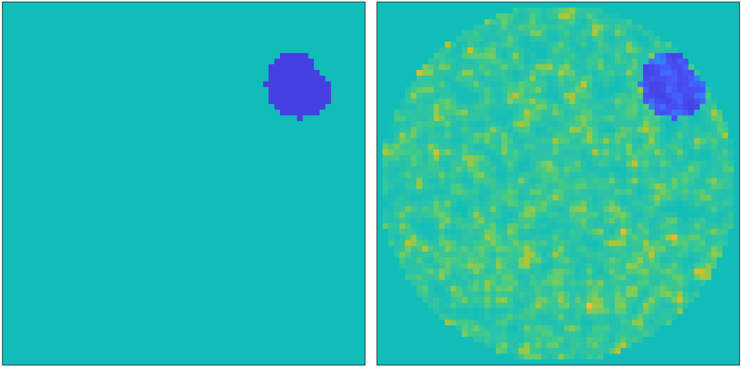
(a) $\Delta\boldsymbol{\sigma}_4$



(b) $\mathbf{h}_g * (\Delta\boldsymbol{\sigma}_4 + \boldsymbol{\delta})$

CNN: Training pair examples (Available at [2])

Denoising and deblurring



(a) $\Delta\boldsymbol{\sigma}_5$ (b) $\Delta\boldsymbol{\sigma}_5 + (\mathbf{h}_g * \boldsymbol{\delta})$

How did we normalize the input of the CNN?

Scale conversion: Let $C = 2.5 * max(abs(\Delta\sigma))$. Then,

Value	**Scale 0-1** [U.A]	**Aux. scale** $[\Omega m]$
Maximum	$\Delta\boldsymbol{\rho}_N = 1$	$\Delta\boldsymbol{\rho}_C = C$
Conductive	$0.5 < \Delta\boldsymbol{\rho}_N < 1$	$0 < \Delta\boldsymbol{\rho}_C < C$
No variation	$\Delta\boldsymbol{\rho}_N = 0.5$	$\Delta\boldsymbol{\rho}_C = 0$
Resistive	$0 < \Delta\boldsymbol{\rho}_N < 0.5$	$-C < \Delta\boldsymbol{\rho}_C < 0$
Minimum	$\Delta\boldsymbol{\rho}_N = 0$	$\Delta\boldsymbol{\rho}_C = -C$

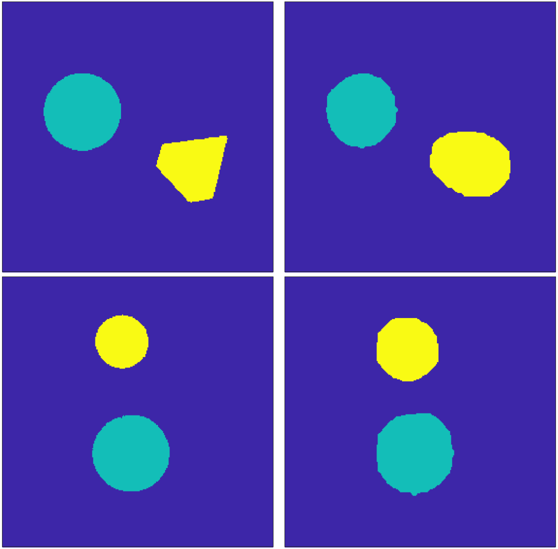
- This choice of C accumulates values closer to 0.5;
- Empirical results show that higher values of C eliminates more information from the image.

Observation: Normalization

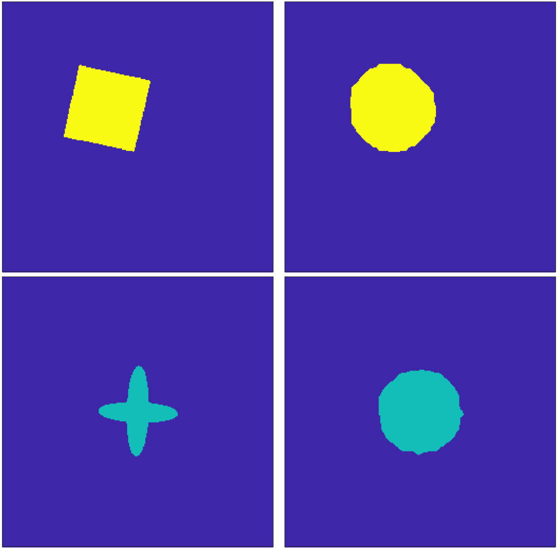
- By the rules, the expected result was the segmented result.
- At each stage of this proposal, it was not necessary to worry about the amplitude of the original conductivity, as we did not have to convert the CNN output back to conductivity units.
- If it was an iterative algorithm, with one application of the CNN per iteration, greater attention would be needed to the signal amplitudes

Proposal results (training set)

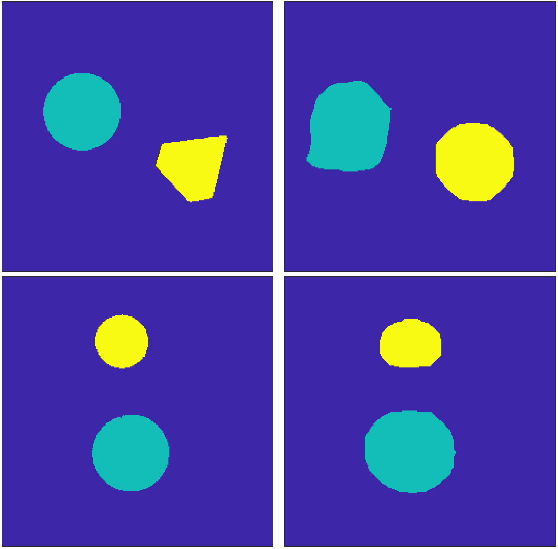
Level 1 (Score = 0.81)



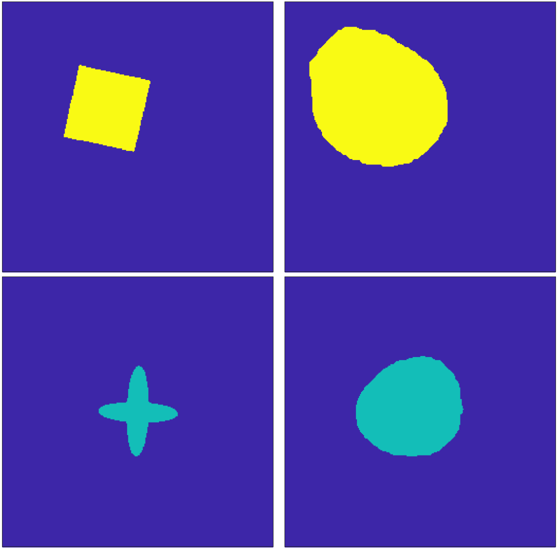
Level 1 (Score = 0.81)



Level 7 (Score = 0.69)



Level 7 (Score $= 0.69$)



Final comments

Proposal drawbacks

- Each result is very dependent on the initial reconstruction
- We didn't correct shape deformation (SSIM < 1)
- CNN training and inference on 64×64 images
- Cases that could cause problems are exactly what we imagine for the higher levels of KTC2023 test set
- With many parameters to adjust, it was not possible to guarantee that the original proposal would be suitable for the test set

Conclusions

- The proposal uses the KTC code base and additional post-processing steps to help segmentation
- It uses a CNN trained with a simulated database consisting of 5000 blurred and noisy circular targets
- Training was fast and only a single training was necessary (same CNN weights at all levels)
- Given the trained CNN, the reconstruction is fast
- In the end, even with circular reconstructed targets, the score did not drop much at the higher levels

[1] M. Räsänen *et al.*
Kuopio tomography challenge 2023 open electrical impedance tomographic dataset (ktc 2023) (Zenodo).
DOI 10.5281/zenodo.8252369, 2024.

[2] R. G. Beraldo *et al.*
Simulated EIT circular targets with noise and blur (Zenodo).
DOI 10.5281/zenodo.10801591, 2024.