Tracin in Semantic Segmentation of Tumor Brains in MRI, an Extended Approach

Tommaso Torda[1,2,†], Simona Gargiulo[1,2,†], Greta Grillo[2,†], Andrea Ciardiello[1,2,†], Cecilia Voena[1,2,†], Stefano Giagu[1,2,†] and Simone Scardapane[1,†]

[1] *"Sapienza" University of Rome, Rome, Italy,*

[2] *National Institute for Nuclear Physics Rome Division, Italy*

Abstract

In recent years, thanks to improved computational power and the availability of big data, AI has become a fundamental tool in basic research and industry. Despite this very rapid development, deep neural networks remain black boxes that are difficult to explain. While a multitude of explainability (xAI) methods have been developed, their effectiveness and usefulness in realistic use cases is understudied. This is a major limitation in the application of these algorithms in sensitive fields such as clinical diagnosis, where the robustness, transparency and reliability of the algorithm are indispensable for its use. In addition, the majority of works have focused on feature attribution (e.g., saliency maps) techniques, neglecting other interesting families of xAI methods such as data influence methods. The aim of this work is to implement, extend and test, for the first time, data influence functions in a challenging clinical problem, namely, the segmentation of tumor brains in Magnetic Resonance Images (MRI). We present a new methodology to calculate an influence score that is generalizable for all semantic segmentation tasks where the different labels are mutually exclusive, which is the standard framework for these tasks.

Keywords

AI, xAI, Deep Learning, Healthcare, Brain Tumors,

1. Introduction

The implementation of Artificial Intelligence (AI) algorithms in the medical domain is continuously increasing, driven by advances in the AI field both from the algorithm and the computational power side, in particular in medical image analysis. Several tasks can nowadays be performed by AI models on medical images, like classification, registration and segmentation.

Medical image segmentation is an essential task for diagnosis, treatment planning and monitoring in the clinical management of many diseases. This task, which consists in the outline of an organ or a lesion in a medical image, is often performed by a clinician (radiologist), but nowadays Deep Neural Networks (DNN), in particular Convolutional Neural Networks (CNN) [1] have proved their effectiveness. While high-performance AI algorithms can be

HC@AIxIA 2023: 2nd AIxIA Workshop on Artificial Intelligence For Healthcare, Rome 2023

[*] Torda Tommaso, PhD

[†] These authors contributed equally.

tommaso.torda@uniroma1.it (T. Torda); simona.gargiulo@uniroma1.it (S. Gargiulo); greta.grillo@unicampus.it (G. Grillo); andrea.ciardiello@uniroma1.it (A. Ciardiello); cecilia.voena@roma1.infn.it (C. Voena); stefano.giagu@uniroma1.it (S. Giagu); simone.scardapane@uniroma1.it (S. Scardapane)

developed, the adoption of AI solutions in the clinical practice is currently strongly limited by lack of trustworthiness due to the little transparency of decisional processes and validation mechanisms of such complex models.

In the medical image analysis domain, there is a wide literature about explainability (xAI) methods with most work related to classification tasks (see [2, 3] for recent and comprehensive reviews). A relevant point that needs to be addressed is how to evaluate the quality of an explanation. Figures of merit have been identified, like for example the ease of use, the plausibility (correctness of the explanation and correspondence to what the user expects), the *faithfulness* (how accurately the explanation reflects the model's true decision process), robustness (effect of changing some aspects of the DNN model) but a standard does not exist yet. One of the challenges here is that the explanation must be helpful for end users, in this case radiologists and clinicians, making this problem interdisciplinary. Some guidelines have been proposed, like the INTRPTR guidelines [4].

In this paper we want to address the issue of xAI in the segmentation task performed by a DNN on multimodal Magnetic Resonance Images (MRI) of brain tumors, one of the leading causes of death worldwide [5].

Since segmentation is a localization problem, the application of visual xAI methods is not obvious, because generated saliency maps, which show the importance of the pixels in the segmentation, is not a useful information alone. An interesting xAI algorithm that has become very popular over the past years is *TracIn* [6], that is never applied to segmentation tasks in medical image imaging so far. *TracIn* belongs to the class of xAI techniques based on approximating the influence a example used in the training process of the network has on the predictions made by the model.

The aim of our work is to implement this technique in a specific clinical problem, the segmentation of tumor brains in multimodal MRI, and to provide information regarding the robustness of the algorithm with respect to different training strategies. To this purpose, the original *Tracin* algorithm is modified since it was originally developed for classification tasks. We consider as reference datasets Brats19 [1], and a standard 2D UNet [7].

2. Material and Methods

2.1. Image dataset

The brain tumor segmentation challenge, BraTS [2], is aimed at evaluating state-of-the-art methods for the segmentation of brain tumors in multimodal MRI. The training dataset for BraTS2019 is composed of 259 cases of high-grade gliomas (**HGG**) and 76 cases of low-grade gliomas (**LGG**), manually annotated by both clinicians and board-certified radiologists. For each patient four MRI scans taken with different modalities are provided: T1, T1Gd, T2, T2-FLAIR. with an image's shape of voxels 240 × 240 × 155. We focused only on HGG patients, dividing the dataset into 207 train patients and 52 validation patients. In the manual label of BraTS19 four classes are provided: the GD-enhancing tumor (**ET - label 4**), the peritumoral edema (**ED - label 2**),

[1] https://www.med.upenn.edu/cbica/brats2019/data.html

[2] http://braintumorsegmentation.org/

the necrotic and non-enhancing tumor core (**NCR/NET - label 1**) and the background (**BKG - label 0**). In the following, we will refer to the GD-enhancing tumor (ET) as label 3 instead of label 4. Each pixel is exclusively assigned to one of these classes.

2.2. Segmentation algorithm

To solve the segmentation task, we chose a popular and well-established neural network, the UNet for 2D segmentation [7], it is easy to implement and has extensive literature.
For the 2D UNet we considered separately the 155 slices along the longitudinal axis. We then took only the central volume of the brain, reducing the number of slices to 10, because influence xAI methods are quite computationally expensive, and we want to reduce the computational time. We also cropped the image on the x-y axes from 240 to 192 pixels. We applied the following data augmentation transformations: elastic transformation, random crop and mirroring with probability of 50%, and then we normalized the intensity between $[-1, 1]$.
As activation function, we used the softmax function. Furthermore, we choose the mean of the Dice Coefficients (D) as loss function along each class. The D is defined for each class as

$$\mathrm{D} = \frac{2(P \cap GT)}{P \cup GT}. \tag{1}$$

This metric measures the overlap between the prediction mask (P) and ground truth (GT) and it is necessary when the region of interest is smaller than the background area. The implementation from the computational point of view is done using the Soft Dice defined as

$$\mathrm{D} = \frac{2\sum P * GT}{\sum P^2 + \sum GT^2 + \epsilon}, \tag{2}$$

where the sum is made over all the pixels of the mask, the product is made pixel by pixel, and ϵ is a small arbitrary parameter in order to avoid NaN values. The Dice score for each separate label is also used as a metric in order to evaluate the trained network performances. The complete loss function than became

$$\mathscr{L} = 1 - \frac{1}{3} \sum_{i \in [1,3]} \mathrm{D}_i, \tag{3}$$

Where i is the class index, and we excluded the background from the loss function calculation. We used Adam optimizer with learning rate of 1e-4 and weight decay of 1e-5 for the first 5 epochs, we switch on stochastic gradient descent (SGD) for the remain epochs. In total we trained the model for 30 epochs, with the train batch size equal to 10. Results of the training process are provided in Table 1.

	NCR/NET	ED	ET
Train	0.90	0.93	0.93
Val	0.77	0.87	0.87

Table 1
Results of training process on the BraTs19 dataset. We report the dice score for the different tumor classes.

2.3. Explainability algorithms

2.3.1. TracIn

The idea of influence based xAI methods is to estimate the effect of removing an example train $\bar{z}$ from the dataset on the loss function $\mathscr{L}$. Pruthi et al. [6], implemented an influence function that monitors loss changes during training, and involves only first order derivative of $\mathscr{L}$. The final equation they obtained, assuming stochastic gradient descent, i.e $\theta_{t+1} = \theta_t - \eta_t \nabla \mathscr{L}(\theta_t , z)$, where θ_t are the parameters of the network at the epoch t, is

$$\text{Tracin}(z, z') = \frac{1}{b} \sum_{t \in \mathscr{C}} \eta_t \nabla \mathscr{L}(\theta_t , z') \cdot \nabla \mathscr{L}(\theta_t , z) \tag{4}$$

where $\mathscr{C}$ are checkpoints, z and z' a train and a test example respectively, b the batch size and η_t the learning rate. We call opponent an example that has a negative value of influence score, and proponent an example that has a positive value of influence score. We chose the checkpoints after the transition to SGD, we considered the first 10 epochs for the calculation of *TracIn*.

2.3.2. Extended methodology for segmentation

Originally, *TracIn* was proposed for classification tasks now we generalize it for segmentation task.

We first consider $\mathscr{L}$ defined in (3). When we evaluate the scalar product between the gradient of $\mathscr{L}$ as (4), we obtain

$$\text{Tracin}(z, z') \approx \sum_{(i,j) \in [1,3]} \nabla D_i(z) \cdot \nabla D_j(z'), \tag{5}$$

where D_i is the dice coefficient corresponding to the class i.

Since we do not want to mix influence contributions for pixels belonging to different classes, we decided to compute the *TracIn* for individual labels.

$$\text{Tracin}(z, z')_{ij} \approx \nabla D_i(z) \cdot \nabla D_j(z'), \tag{6}$$

in this way we get 3x3 matrices of *TracIn*, for $i = j$ we compute influence score between same classes for the treat examples, and for $i \neq j$ the influence between different classes. A precondition that we must meet is that the segmentation's classes are mutually exclusive. However, this is not a strong constraint because it is the standard framework for this type of task.

Also, we consider for each class only regions with NN output over a certain threshold (0.8). This is to further reduce the contribution of averaging heterogeneous pixels together.

2.4. Consistency tests

After defining a methodology for calculating *TracIn* for segmentation task, some consistency tests are carried out to verify the goodness of the algorithm.

2.4.1. Self-influence

One of the main check to show in the case of *TracIn* is the self-influence matrix of the train dataset against itself. Self-influence is a matrix where *TracIn* scores of train examples against themselves are reported. We defined the normalized self-influence as

$$SI(x,y)_{ij} = \frac{Tracin(x,y)_{ij}}{\sqrt{Tracin(x,x)_{ii} \times Tracin(y,y)_{jj}}}. \tag{7}$$

Where $SI(x,y)_{ij}$ and $Tracin(x,y)_{ij}$ are respectively the self-influence and the *TracIn* score between pair of train examples x and y for Dice i and j.
We expect that each train example is the main proponent of itself thus higher value of SI on the diagonal of the self-influence matrix.

2.4.2. Robustness test

Robustness tests are proposed to check the stability of xAI's method with respect to small variations in both statistics and training strategy [8].

We have several metrics that we can adopt to study the robustness of the algorithm. In general, let us call $e_i(x)$ the i-th explanation for the example x, then we can test the Robustness (R) of the explanation as

$$R = \frac{1}{N} \sum_{i \neq j} s[e_i(x), e_j(x)] \tag{8}$$

where N is a normalization, and s can be any similarity metrics, in our case we chose the cosine similarity $s(a,b) = \frac{a^T b}{\|a\|_2 \cdot \|b\|_2}$. In our case $e_i(x) = Tracin(x,z)$ where z is the train dataset.

Statistical robustness

We repeat the training of the network 3 times, leaving parameters and hyperparameters unchanged. We produce 3 explanation vector $e_i(x)$ for the same dataset, and using (8) we evaluate the statistical robustness of *TracIn*.

Small dataset variation robustness

We use the method of k-fold cross-validation. We divide the dataset into 3 groups, as shown in the Figure 1. Then we produced the explanation vector $e_i(x)$, for the 3 different training. Applying (8), only on the intersection between pair of explanation $e_i(x), e_j(x)$, we evaluate the robustness of *TracIn* respect of small variation on the dataset.

Transformations robustness Take the neural network that is invariant with respect to the symmetry group $\mathcal{G}$ and an explanation $e(x)$. We can measure the invariance of $e(x)$ under $\mathcal{G}$ using (8). Assume that the element on the dataset x, transforms under the application of $\mathcal{G}$ as $x' = \rho(g)x$ where $\rho(g)$ is a representation of the element $g \in \mathcal{G}$. If $e(\rho(g)x) \approx e(x)$, than e is invariant.

We tested *TracIn* invariance using 2 symmetries that are commonly used in training neural networks in biomedical tasks: elastic deformation and mirroring.

After training the neural network, we produce an explanation for each transformation by taking the train dataset and applying those transformations on it. Then we evaluate the robustness using (8).

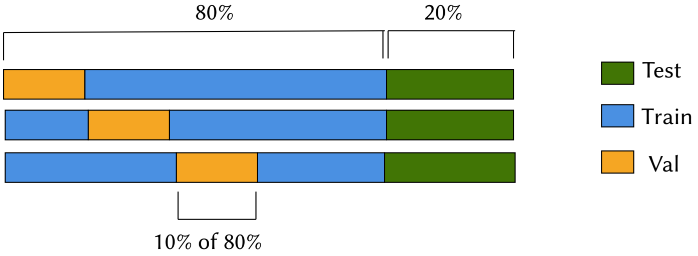

Figure 1: K-fold cross-validation for test the robustness of *TracIn* respect a small variation of 10% on the dataset.

3. Results

Self-influence

The normalized self-influence defined in (7), collects a lot of important information, first we expect the matrix to have a bright diagonal, as each example must be a strong proponent for the prediction of itself.

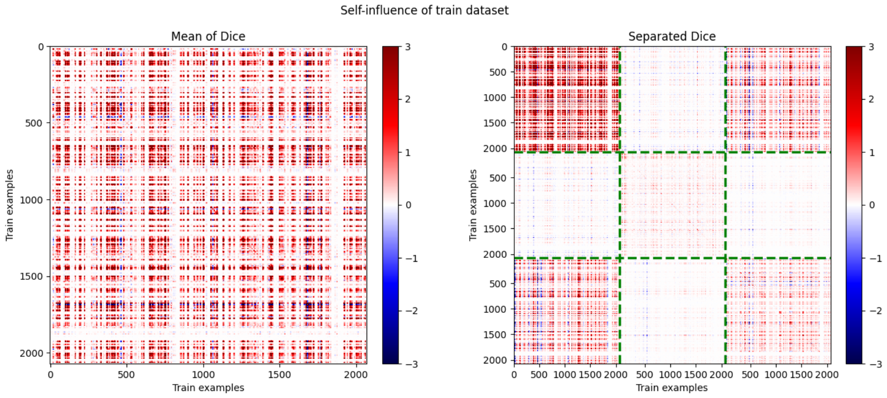

Figure 2: Normalized self-influence of train dataset. The left one is the influence matrix using the complete loss function, and sorted so as to have slices of the same patient close together. The right one is the influence between all the Dice pair $D_i \cdot D_j$, Where the first block is $D_1 \cdot D_1$ with examples ordered like above, the second block is $D_1 \cdot D_2$, and so on.

The two Figures in 2 are obtained by ordering the train examples in such a way that each slice of the same patient is close to the others. The figure on the left is obtained using the original *TracIn* calculation (Eq. 5), where the full loss function is considered, i.e., the sum over all classes. What we expect to see is a 10 × 10 block matrix (10 are the slices we consider of the single patient). The influence matrix we observe present a bright diagonal, but there are contributions of the same order of magnitude among all other patients. This means that on average, any patient contributes equally on the prediction of the others.

The figure on the right is instead obtained by implementing our methodology (Eq. 6). In this case the first block of 2070 × 2070 influences corresponds to pixels belonging to label 1 of patients sorted as described above, the second block will be the mixed influences between labels 1 and 2 of the same list of patients, and so on. The clustering that emerges is not sensitive to local differences between patients, but only to global differences between classes. In fact, as we can see, in the blocks along the diagonal corresponding to the *TracIn* evaluated among the same tumor classes, we have stereotyped matrices, which exhibit homogeneous influence by modulus and sign regardless of the example chosen. Going outside the diagonal, the modulus of influence decreases, indicating that different regions have lower influence among them. However we have noise between label 1 and 3, this means that the predictions on these regions are not completely de-correlated with each other.

Robustness test We first checked the statistical correlation between the influence vectors produced by repeating the training and keeping the same parameters and hyperparameters. The average correlation for the individual Dices is then used as normalization for K-fold cross-validation.

Significant statistical variation is observed for label 3 in each test in Figure 3. A small variation of 10% on the dataset has comparable effects on the robustness of the explanation. However, what we notice is that *TracIn* is not robust to repeated trains (Table 3).

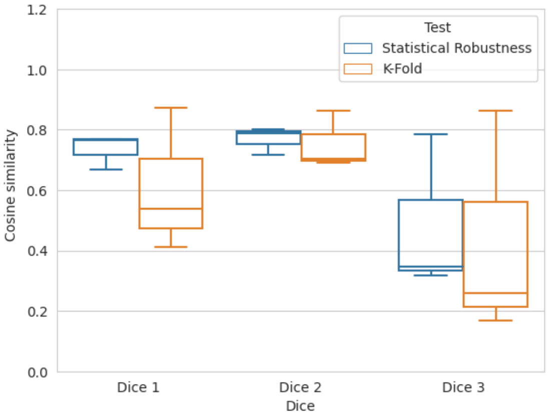

Figure 3: Statistical and K-fold Tests. Each box-plot is produced by evaluating the robustness metrics along the test examples.

	Statistical Robustness	K-Fold	Normalized K-fold
Cosine similarity for Dice 1	0.74 ±0.08	0.61 ±0.22	0.82
Cosine similarity for Dice 2	0.77 ±0.06	0.75 ±0.10	0.97
Cosine similarity for Dice 3	0.49 ±0.23	0.43 ±0.28	0.88

Table 2
Table of results for the Statistical and K-fold test For robustness *TracIn*. Mean and standard deviation are reported.

This might seem surprising, but it is related to what was seen above in the case of the self-influence matrix. Under the assumption that *TracIn* is a $faithful$ explanation of the network decision model, the fact that the explanation vector is not robust for repeated trains has a direct interpretation. At each training the network is initialized in a random parameter space, the train of the network occurs stochastically (the batches and optimization change each time), and, at the end of each training cycle, we end up in a different, and hopefully, equivalent minimum in the landscape of the loss function. As we saw for the case of the self-inference matrix, examples belonging to the same class have stereotypical influence. This means that each prediction in the network is homogeneously influenced by any other example we have used, since they have a comparable amount of information that can be extracted. The non robustness of the explanation means that we are looking at contour lines in the loss function landscape that are essentially flat, where each trajectory in this framework is analogous to all others. Thus, the explanation vector changes with each iteration, having no more important examples of trains than others for reaching the optimal minimum during the minimization process. What remains truly robust is the influence the different classes have on each other as it can be seen in Figure 4.

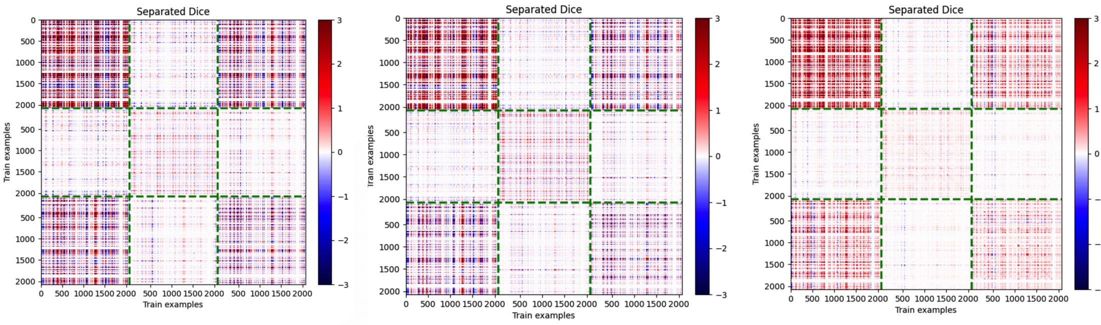

Figure 4: Three different versions of the self-influence matrix. From left to right, we repeated the training two times; the last matrix on the right is obtained by considering only the last 10 checkpoints of the same training of the middle one. In all three cases we can see that the overall structure of influence between classes remains robust.

Regarding the invariance of the *TracIn*, we observe that when the neural network is invariant with respect to a $\mathscr{G}$ group, the *TracIn* also resepect such invariance (Figure 5 and Table 3). The results are in agreement with what was previously observed in [8].

	Cosine similarity
Original vs Elastic deformation	0.93 ±0.18
Original vs Mirroring	0.95 ±0.12

Table 3
Table of results for the Invariance of *TracIn* under popular transformations. Mean and standard deviation are reported.

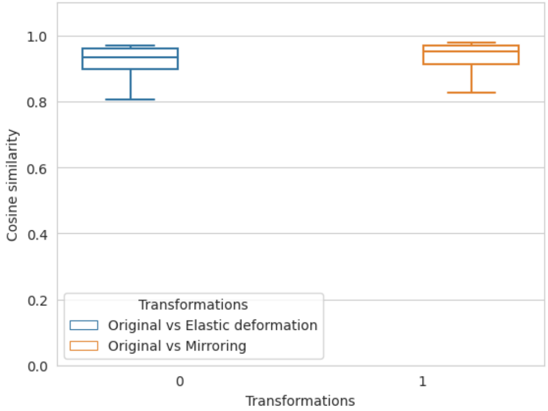

Figure 5: Invariance Test. Each box-plot is produced by evaluating the robustness metrics along the test examples.

4. Conclusion

In this paper we proposed an extended methodology to implement *TracIn* in a specific problem, brain tumor multiclass MRI segmentation. However, this is generalizable to all segmentation problems where classes are mutually exclusive. We found that an explanation cannot be given between examples (slices) of the same class, but only by separating the analysis by distinct classes and reducing the number of pixels on each examples. This is similar to the information you can gather with a saliency map, where the most influential pixels for predicting a class are those belonging to the same one.

We then analyzed the robustness of the algorithm in several frameworks. First by re-training the network leaving the parameters and hyperparameters unchanged, then by changing the 10% of the dataset. In this case we concluded that *TracIn* is not robust for different trains. Under the assumption that *TracIn* is a $faithful$ explanation of the network decision model, what we are observing is a flat loss landscape, where several trajectories turn out to be equivalent for reaching the minimum. What really remains informative then are the differences between different classes, but because of the nature of segmentation, differences on the same classes not only become irrelevant but in general, even for different tasks, are not robust. Leaving the network unchanged, and instead studying the robustness toward certain transformations, we verified the invariance of *TracIn* respect these.

Like other post-hoc visual explainability techniques, *TracIn* also suffers from a difficulty in interpreting its results.

Future work should focus not only on giving an explanation on the basis of influences but also on giving information regarding the *faithfulness* of the algorithm.

References

[1] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, 2014. arXiv:1311.2524.

[2] B. H. van der Velden, H. J. Kuijf, K. G. Gilhuijs, M. A. Viergever, Explainable artificial intelligence (XAI) in deep learning-based medical image analysis, Medical Image Analysis 79 (2022) 102470. URL: https://doi.org/10.1016%2Fj.media.2022.102470. doi:10.1016/j.media.2022.102470.

[3] K. Borys, Y. A. Schmitt, M. Nauta, C. Seifert, N. Krämer, C. M. Friedrich, F. Nensa, Explainable ai in medical imaging: An overview for clinical practitioners – beyond saliency-based xai approaches, European Journal of Radiology 162 (2023) 110786. URL: https://www.sciencedirect.com/science/article/pii/S0720048X23001006. doi:https://doi.org/10.1016/j.ejrad.2023.110786.

[4] H. Chen, C. Gomez, C.-M. Huang, M. Unberath, Explainable medical imaging ai needs human-centered design: Guidelines and evidence from a systematic review, 2022. arXiv:2112.12596.

[5] A. Wadhwa, A. Bhardwaj, V. Singh Verma, A review on brain tumor segmentation of mri images, Magnetic Resonance Imaging 61 (2019) 247–259. URL: https://www.sciencedirect.com/science/article/pii/S0730725X19300347. doi:https://doi.org/10.1016/j.mri.2019.05.043.

[6] G. Pruthi, F. Liu, M. Sundararajan, S. Kale, Estimating training data influence by tracing gradient descent, NeurIPS 2020 (2020). arXiv:2002.08484.

[7] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, 2015. arXiv:1505.04597.

[8] J. Crabbé, M. van der Schaar, Evaluating the robustness of interpretability methods through explanation invariance and equivariance, 2023. arXiv:2304.06715.