

Interactive Visualization of 3D CNN Relevance Maps to Aid Model Comprehensibility

Application to the Detection of Alzheimer’s Disease in MRI Images

Martin Dyrba¹, Moritz Hanzig^{1,2}

¹German Center for Neurodegenerative Diseases (DZNE), Rostock, Germany

²Institute of Visual & Analytic Computing, University of Rostock, Germany
`martin.dyrba@dzne.de`

Abstract. Relevance maps derived from convolutional neural networks (CNN) indicate the influence of a particular image region on the decision of the CNN model. Individual maps are obtained for each single input 3D MRI image and various visualization options need to be adjusted to improve information content. In the use case of model prototyping and comparison, the common approach to save the 3D relevance maps to disk is impractical given the large number of combinations. Therefore, we developed a web application to aid interactive inspection of CNN relevance maps. For the requirements analysis, we interviewed several people from different stakeholder groups (model/visualization developers, radiology/neurology staff) following a participatory design approach. The visualization software was conceptually designed in a Model–View–Controller paradigm and implemented using the Python visualization library Bokeh. This framework allowed a Python server back-end directly executing the CNN model and related code, and a HTML/Javascript front-end running in any web browser. Slice-based 2D views were realized for each axis, accompanied by several visual guides to improve usability and quick navigation to image areas with high relevance. The interactive visualization tool greatly improved model inspection and comparison for developers. Owing to the well-structured implementation, it can be easily adapted to other CNN models and types of input data.

1 Introduction

Convolutional neural networks (CNN) achieved a high accuracy for the automated detection of disease patterns in MRI scans. Several relevance mapping algorithms have been proposed to generate heatmaps that indicate the influence of a particular image region on the decision of the CNN model [1,2]. Two previous studies compared CNN relevance mapping algorithms with respect to brain regions driving the detection of Alzheimer’s disease in structural T1-weighted MRI [3,4]. These relevance maps were found to greatly improve CNN comprehensibility and identification of reasons why a model failed [1,3]. Notably, these

approaches generate 3D relevance maps for each single input image (=MRI scan). In addition, several post-processing steps are required in order to improve their visual appeal and information content. These steps include smoothing, color scale transformation, relevance score and cluster size thresholding, which are not implemented in the feature portfolio of common MRI viewers. Further, preparation of static images from a specific parameter set yields large amounts of output files, which is prone to losing track of the particular parameter settings used to generating these files in explorative research with various CNN model and post-processing parameter combinations.

In this paper, we present an interactive visualization toolkit for the online generation, parameterization, and inspection of CNN relevance maps for individual MRI scans. In the following sections, we describe the conceptual considerations and implementation, followed by a demonstration of the realized user interface and use case.

2 Materials and methods

We used pretrained CNN models obtained from [4,6], which were implemented in Keras 2.2.4 and Tensorflow 1.15. The CNN visualization library iNNvestigate 1.0.8 [2] was used to derive the relevance maps. After drafting a first prototype user interface for the visualization, we collected a list of key requirements from a range of stakeholders following a participatory design approach. Therefore, we interviewed two physicians trained in radiology/neurology, two experienced visualization developers, and two machine learning model developers. From their comments, we defined the list of requirements:

- Directly run in CNN modeling environment (Python)
- Optional: remote display for the case where data handling and model execution need to be run remotely
- Visualization as slice-based 2D plots, which clinical users are familiar with
- For regular users: interactive selection of MRI scans, adjustable relevance and cluster size thresholds
- For expert users: selection of alternative CNN models and relevance mapping algorithms

The Python visualization library Bokeh [5] met the requirements with respect to Python runtime environment and remote viewing instance in a web browser. It provides a Python server instance back-end and Javascript browser libraries front-end to remotely trigger Python function calls and return execution results to the web browser for displaying.

We divided the implementation into three components following the well-established Model-View-Controller design pattern. Fig. 1 provides an overview of implemented methods and Fig. 2 shows a sequence diagram of function calls being executed when selecting a new MRI scan. In addition to the key requirements, we implemented various visual guides in order to facilitate parameterization and quick navigation to brain regions with high relevance scores (see Fig. 3).

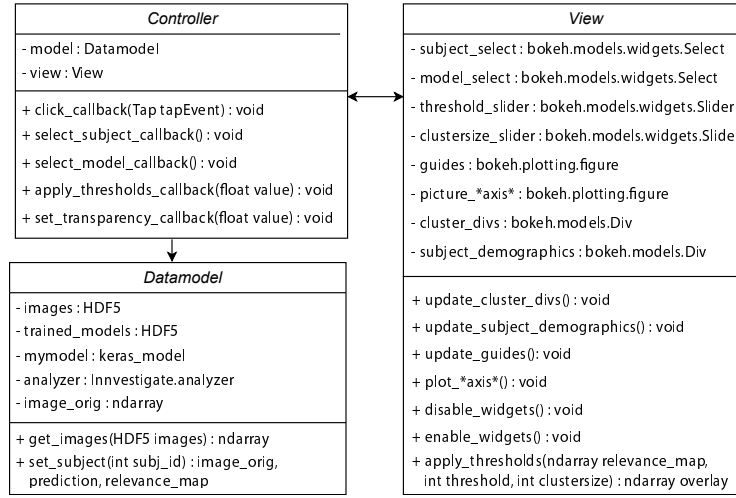


Fig. 1. Class diagram illustrating core components and functions.

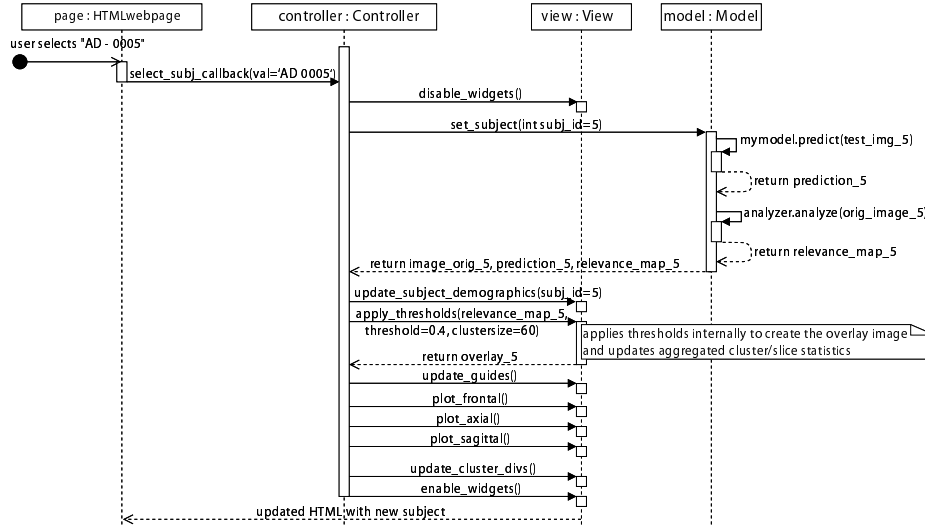


Fig. 2. Sequence diagram of functions being executed when loading a new MRI scan.

Among them are (a) a histogram providing the distribution of cluster sizes next to the cluster size threshold slider, (b) plots visualizing the amount of positive and negative relevance per slice next to the slice selection sliders, and (c) statistical information on the currently selected cluster. Further, assuming spatially normalized MRI data in MNI reference space, we added (d) atlas-based anatomical region lookup for the current cursor/cross-hair position and (e) the option to display the outline of the anatomical region to simplify visual comparison with the cluster location.

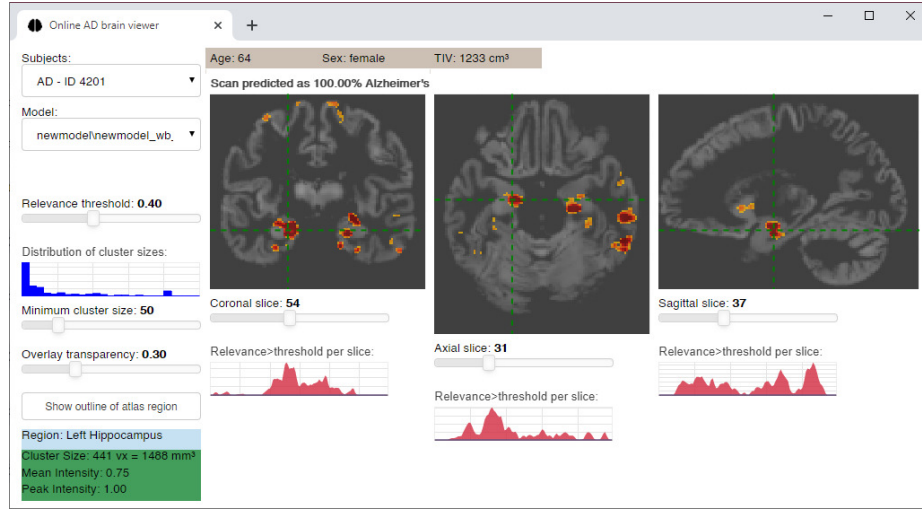


Fig. 3. Interactive web application user interface for 3D CNN relevance map visualization for a patient with dementia due to Alzheimer’s disease.

3 Results

The realized user interface is shown in Fig. 3. The source code is freely available on GitHub: <https://github.com/martindyrba/DeepLearningInteractiveVis>. The employed relevance mapping algorithm was initially fixed to layer-wise relevance propagation (LRP) as this method was already applied previously [3,4]. The distribution and location of clusters with highest relevance scores varied between people with most consistent contributions from hippocampus, putamen and thalamus (Fig. 4). Notably, the highlighted regions mostly indicated actual gray matter atrophy as visible from the background images. This was confirmed by a quantitative comparison in which automatically derived hippocampus volume measures highly correlated with the aggregated relevance scores in the hippocampus region (Pearson’s $r \approx -0.81$, see [6] for further details).

The used high-level programming interfaces of Keras, iNNvestigate, and Bokeh enabled a clean and structured programming of the web application. The complete application was realized in approximately 700 lines of code including view specification and program logic (model, controller). The app does not require a GPU to be available on the host, i.e. works smoothly on CPU. Loading of a new person currently takes ≈ 5 sec and loading a new model takes ≈ 15 sec. Adjusting the other sliders or directly clicking on the brain image/clusters updates the visualizations with a short latency of ≈ 200 ms. With the web page front-end, the visualization can also be run remotely on mobile devices such as tablets or smartphones without modification.

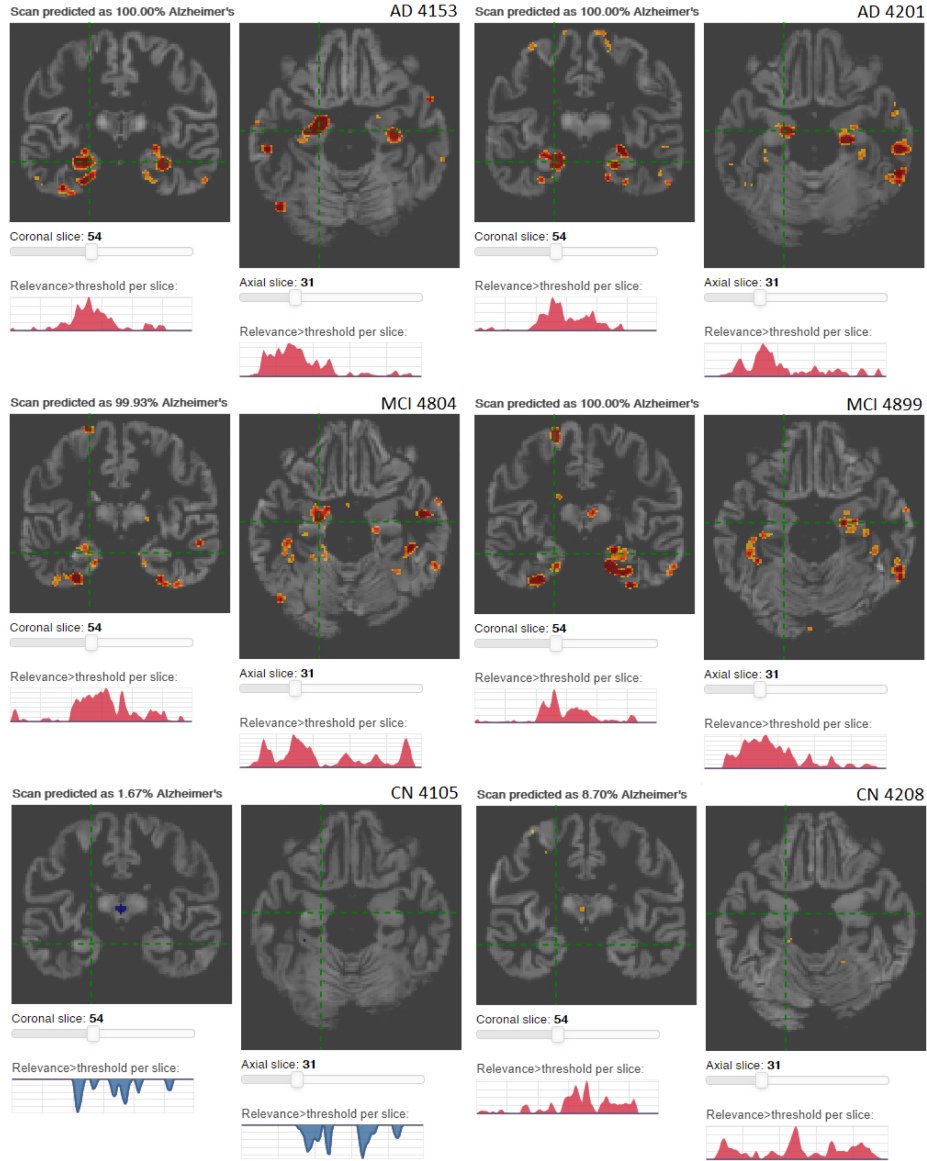


Fig. 4. Comparison of 3D CNN relevance maps for six randomly selected MRI scans. Top row: two patients with dementia due to Alzheimer's disease, middle row: two patients with mild cognitive impairment, bottom row: two controls with normal cognition.

4 Discussion

The presented visualization framework allows the inspection of CNN relevance maps for individuals and the assessment of drivers of the CNN decision. Therefore, it contributed to model comprehensibility in the sense that it revealed the

regional sensitivity of the CNN models, which is currently not assessed in the majority of papers with focus on model accuracy. As previously reported for other application domains [1], we found an association of model performance and relevant regions, which means that less accurate models mainly considered brain regions of low clinical relevance for AD. Thus, relevance maps provide a useful tool for CNN model 'debugging'.

Consulting experts from various disciplines for the requirements analysis greatly improved usability of the initial prototype application with many useful comments and recommendations such as adding the visual guides. The presented application was implemented as model- and data-agnostic tool such that it can be easily adjusted for other types of 3D input data.

The interactive web application met all the requirements defined initially (see section 2 above). A clear disadvantage is the small latency in reactivity causing a short delay of navigation actions. This is due to the data handling on the server and transfer of the relevance map slices to the client as byte stream, which could only be circumvented if both data model and viewer components run on the same system as native Python application.

For future work, there are ideas on additional features, for instance the import of new MRI scans, export for relevance maps, and 3D rendering view components. Further evaluation of the actual improvement of CNN comprehensibility for end users such as clinical staff is advised. Most importantly, more research is required to define evaluation metrics for relevance map quality and plausibility.

In summary, we presented a concept and implementation of an interactive online visualization application to inspect 3D CNN relevance maps and adjust display parameters as appropriate. Highlighting the individual's image regions with highest contribution on the particular decision of the CNN model in a simple and intuitive way makes this tool greatly enhancing model inspection and comparison for developers.

References

1. Samek W, Binder A, Montavon G, et al. Evaluating the Visualization of What a Deep Neural Network Has Learned. *IEEE T Neur Net Lear*. 2017;28(11):2660–2673.
2. Alber M, Lapuschkin S, Seegerer P, et al. iNNvestigate neural networks! *J Mach Learn Res*. 2019;20:1–8.
3. Böhle M, Eitel F, Weygandt M, et al. Layer-Wise Relevance Propagation for Explaining Deep Neural Network Decisions in MRI-Based Alzheimer's Disease Classification. *Front Aging Neurosci*. 2019;11:194.
4. Dyrba M, Pallath AH, Marzban EN. Comparison of CNN Visualization Methods to Aid Model Interpretability for Detecting Alzheimer's Disease. *Proc BVM*. 2020; p. 307–312.
5. Bokeh contributors. The Bokeh Visualization Library; 2020. <https://bokeh.org>.
6. Dyrba M, Hanzig M, Altenstein S, et al.. Improving 3D convolutional neural network comprehensibility via interactive visualization of relevance maps: Evaluation in Alzheimer's disease; 2020. <https://arxiv.org/abs/2012.10294>.