Contents lists available at ScienceDirect

SoftwareX

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

Original software publication

Open-source skull reconstruction with MONAI[☆]

Jianning Li[a,d,e,*], André Ferreira[a,d,i], Behrus Puladi[f,l], Victor Alves[i], Michael Kamp[b,j,k], Moon Kim[a,b,g], Felix Nensa[a,g], Jens Kleesiek[a,b,c], Seyed-Ahmad Ahmadi[h], Jan Egger[a,b,d,e,*]

[a] *Institute for Artificial Intelligence in Medicine (IKIM), Essen University Hospital (AöR), Girardetstraße 2, 45131 Essen, Germany*
[b] *Cancer Research Center Cologne Essen (CCCE), University Medicine Essen (AöR), Hufelandstraße 55, 45147 Essen, Germany*
[c] *German Cancer Consortium (DKTK), Partner Site Essen, Hufelandstraße 55, 45147 Essen, Germany*
[d] *Computer Algorithms for Medicine Laboratory (Cafe), Graz, Austria*
[e] *Institute of Computer Graphics and Vision (ICG), Graz University of Technology, Inffeldgasse 16c, 8010 Graz, Austria*
[f] *Department of Oral and Maxillofacial Surgery, University Hospital RWTH Aachen, Pauwelsstraße 30, 52074 Aachen, Germany*
[g] *Institute of Diagnostic and Interventional Radiology and Neuroradiology, University Hospital Essen (AöR), Essen, Germany*
[h] *NVIDIA GmbH, Bavaria Towers - Blue Tower, Einsteinstrasse 172, 81677 Munich, Germany*
[i] *ALGORITMI Research Centre/LASI, University of Minho, Braga, Portugal*
[j] *Institute for Neuroinformatics, Ruhr University Bochum, Germany*
[k] *Data Science and AI Department, Monash University, Melbourne, Australia*
[l] *Institute of Medical Informatics, University Hospital RWTH Aachen, Pauwelsstraße 30, 52074 Aachen, Germany*

ARTICLE INFO

Article history:
Received 7 October 2022
Received in revised form 3 May 2023
Accepted 1 June 2023

Keywords:
Skull reconstruction
Research contribution
MONAI
Open-source
API
PyTorch
Python
Deep learning
Pre-trained model
Cranial implant design
Cranioplasty
Craniotomy
Craniectomy
CT
Bone
Head
Face

ABSTRACT

We present a deep learning model based on an autoencoder for the reconstruction of cranial and facial defects using the Medical Open Network for Artificial Intelligence (MONAI) framework, which has been pre-trained on the MUG500+ and SkullFix dataset. The implementation follows the MONAI contribution guidelines, hence, it can be easily tried out and used, and extended by MONAI users. The primary goal of this paper lies in the investigation of open-sourcing codes and pre-trained deep learning models under the MONAI framework. The pre-trained models generated in this work deliver reasonable results on the cranial and facial reconstruction task and provide an ideal starting-point for other researchers interested in further investigating the topic. We released the codes and the pre-trained model at the official MONAI 'research contributions' GitHub repository: https://github.com/Project-MONAI/research-contributions/tree/master/SkullRec. This contribution has two novelties: 1. Pre-training an autoencoder on the MUG500+ and SkullFix dataset for cranial and facial reconstruction using MONAI, and open-sourcing the codes and weights for other MONAI users; 2. Demonstrating that existing MONAI tutorials can be easily adapted to new use cases, such as skull (cranial and facial) reconstruction.

[☆] We acknowledge CAMed (COMET K-Project 871132), FWF enFaced (KLI 678), FWF enFaced 2.0 (KLI 1044) and KITE (Plattform für KI-Translation Essen) from the REACT-EU initiative (https://kite.ikim.nrw/, EFRE-0801977). Further, we acknowledge NVIDIA, United States for the support.

[*] Corresponding author at: Institute for Artificial Intelligence in Medicine (IKIM), Essen University Hospital (AöR), Girardetstraße 2, 45131 Essen, Germany.

E-mail addresses: Jianning.Li@uk-essen.de (Jianning Li), Jan.Egger@uk-essen.de (Jan Egger).

https://doi.org/10.1016/j.softx.2023.101432

Code metadata

Current code version	1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-22-00320
Permanent link to Reproducible Capsule	
Legal Code License	CC-BY-NC-SA
Code versioning system used	none
Software code languages, tools, and services used	python
Compilation requirements, operating environments and dependencies	monai version: 0.8.1, pytorch version: 1.11.0
If available Link to developer documentation/manual	none
Support email for questions	jianning.li@uk-essen.de

1. Introduction

Nowadays, open-sourcing software, especially (pre-trained) deep learning models, has become increasingly important. Over the years, medical image analysis experienced a tremendous transformation. Over a decade ago, algorithms had to be implemented and optimized with low-level programming languages, like C or C++, to run in a reasonable time on a desktop PC, which was not as powerful as today's computers. Nowadays, users have high-level scripting languages like Python, and frameworks like PyTorch and TensorFlow, along with a sea of public code repositories at hand. As a result, implementations that had thousands of lines of C or C++ code in the past, can now be scripted with a few lines and in addition executed in a fraction of the time. To put this even on a higher level, the Medical Open Network for Artificial Intelligence (MONAI) framework (https://monai.io/ [1]) tailors medical imaging research to an even more convenient process, which can boost and push the whole field. The MONAI framework is a freely available, community-supported, open-source and PyTorch-based framework, that also allows pre-trained models to be integrated as research contributions. To put these claims to the acid test, especially, the point of Easy Integration, we investigate the use of MONAI for open-sourcing pre-trained deep learning models. In particular, we constructed an autoencoder using the MONAI framework, and trained the network for skull reconstruction on the MUG500+ and SkullFix dataset [2,3]. The network is trained for reconstructing a complete skull given as input an incomplete skull with cranial or facial defect. We then pushed the codes and the pre-trained autoencoder to the MONAI GitHub repository as a research contribution at https://github.com/Project-MONAI/research-contributions.

In our experience, research studies that provide codes and pre-trained models for reproducing the reported results tend to attract much more attention (and therefore citations) than those that do not, benefiting not only the original authors but also other researchers working on the same topic. To our knowledge, deep learning-based craniofacial reconstruction has only recently become known to a small community working on automatic craniofacial implant design, and there still lacks an open-source deep learning library dedicated to craniofacial reconstruction. We believe that integrating such a tool into MONAI, which has already established a decent user base in medical imaging, will broadcast the problem to a wider audience and potentially benefit those willing to further contribute to the topic.

Project MONAI was initiated jointly by NVIDIA and King's College London, aiming to create an inclusive and community-supported platform, where AI researchers can exchange the best practices of artificial intelligence in healthcare across academia and industry. Among other tools (e.g., MONAI Label), project MONAI created the MONAI framework, which is an open-source and freely available deep learning library that specifically aims at healthcare imaging. It has already been used by researchers, like [4–6]. MONAI provides domain-optimized foundational capabilities for developing healthcare imaging training workflows in a native PyTorch paradigm. In doing so, MONAI features (according to their website):

- Open Source Design: MONAI is an open-source project. It is built on top of PyTorch and is released under the Apache 2.0 license.
- Standardization: Aiming to capture best practices of AI development for healthcare researchers, with an immediate focus on medical imaging.
- User Friendly API: Providing user-comprehensible error messages and easy to program API interfaces.
- Reproducibility: Provides reproducibility of research experiments for comparisons against state-of-the-art implementations.
- Easy Integration: Designed to be compatible with existing efforts and ease of 3rd party integration for various components.
- High Quality: Delivering high-quality software with enterprise-grade development, tutorials for getting started and robust validation & documentation.

Existing MONAI research contributions include, for example, DiNTS (Differentiable Neural Network Topology Search for 3D Medical Image Segmentation) [7,8], BTCV (3D multi-organ segmentation with UNEt TRansformers) for the Beyond the Cranial Vault challenge) [9], COPLE-Net (COVID-19 Pneumonia Lesion Segmentation) [10] and LAMP (Large Deep Nets with Automated Model Parallelism for Image Segmentation) [11].

2. Method

This section describes the main workflow of the contribution, including dataset preprocessing, data format conversion, data resizing and model configurations as well as details about model training, implemented within the MONAI framework.

2.1. Dataset and preprocessing

Two datasets are involved in this study. (1) the MUG500+ dataset [2], which contains 500 complete skulls (e.g., first and third column, Fig. 1) in NRRD format. 21 of the image files were found to be corrupted, and were discarded. The remaining 479 complete skulls were split into a training set with 429 skulls and a test set with 50 skulls. Of the 429 complete skulls in the training set, 253 skulls are inserted with cranial defects (e.g., second column, Fig. 1) and 176 skulls with facial defects (e.g., fourth column, Fig. 1). Correspondingly, we created two defective cases (one with cranial defect and one with facial defect) for each of the 50 test skulls, resulting in $50 \times 2 = 100$ defective test skull samples. The size, shape and position of the defects vary on different skulls. A *complete-defect* skull pair comprises of a defective skull (with either a cranial or a facial defect) and the corresponding complete skull. Therefore, the training set contains 429 such *complete-defect* pairs, and the test set contains $50 \times 2 = 100$ *complete-defect* pairs.

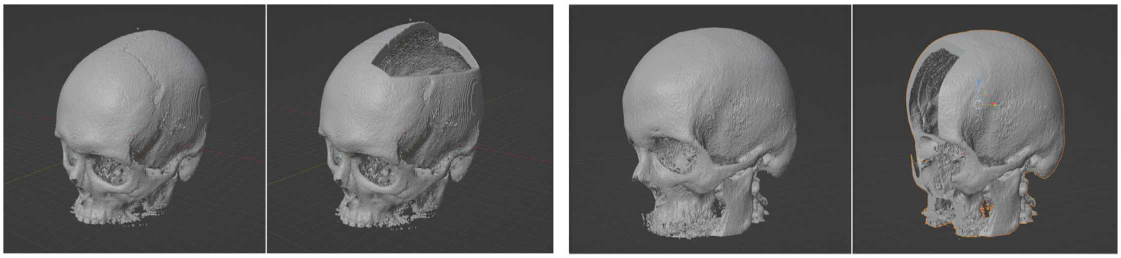

Fig. 1. A complete skull (first, third column) and the corresponding skull with a cranial defect (second column) and a facial defect (fourth column) in the MUG500+ dataset.

The cranial and facial defects were created automatically using the defect creation script in the MONAI *SkullRec* repository. 79 *complete-defect* pairs are further split from the training set for validation during training. The axial dimension of all the skull images were cropped to 256. (2) Besides the MUG500+ dataset, we also trained the MONAI network on the SkullFix dataset [3], which contains 100 *complete-defect* skull pairs for training and 100 pairs for evaluation. In this study, we replaced the original cranial defects with facial defects created the same way as in the MUG500+ dataset, and trained the MONAI network for automatic facial reconstruction.

2.2. Data format conversion

The MONAI framework is optimized for the use of Neuroimaging Informatics Technology Initiative (NIfTI) files due to the integration of the Nibabel [12] library, which makes it easier to use NIfTI files. For this reason, it was necessary to first convert the dataset (which are available in the NRRD file format [13]) to the NIfTI format. Note that even though NIfTI is a common format in medical imaging, MONAI's image loader can also directly work with NRRD files. There are several tools that can perform this conversion. We chose to use the Python library Visualization ToolKit (VTK) to perform this conversion [14]. Then, the MONAI network is used to train the model on the converted dataset. The final output is the trained model, as shown in Fig. 2.

A download link to the converted and processed dataset is also provided in the MONAI SkullRec Github repository. Users can use the dataset directly for training.

2.3. Data resizing and model configurations

The function *Resize* was applied to resize the scans to 256 × 256 × 128 using the interpolation mode *Area*. The intensity range scale was not used, because the datasets are already binary. The model was implemented as a simple version of an autoencoder (AE) from the MONAI framework, with the following configurations: spatial_dims=3; in_channels = 1; out_channels = 2; channels = (32, 64, 64, 128, 128, 256); strides = (2, 2, 2, 2, 2, 2); num_res_units = 0. The AE is symmetrical, i.e. the downsampling has the same number of layers as the upsampling, as illustrated in Fig. 3. Major changes to the MONAI framework were avoided by adapting the work pipeline to facilitate further use by others.

2.4. Model training

The autoencoder model described above is trained using the *DiceLoss* loss function and the *Adam* optimiser with a batch size of two, using respectively the MUG500+ and SkullFix dataset. The function *DiceMetric* is used in the validation step, which is executed every two training epochs. The model is trained for 80 epochs in total, but only the checkpoint corresponding to the best validation score is saved at the end of training. However, it is important to note that, sometimes the validation score can be unreliable, if the validation set is not representative of the dataset. Therefore, it is advised to save not only the best validation checkpoint but also the checkpoint at the last training epoch. When training on the SkullFix dataset, the last checkpoint at the end of training is saved, since SkullFix does not contain a validation set.

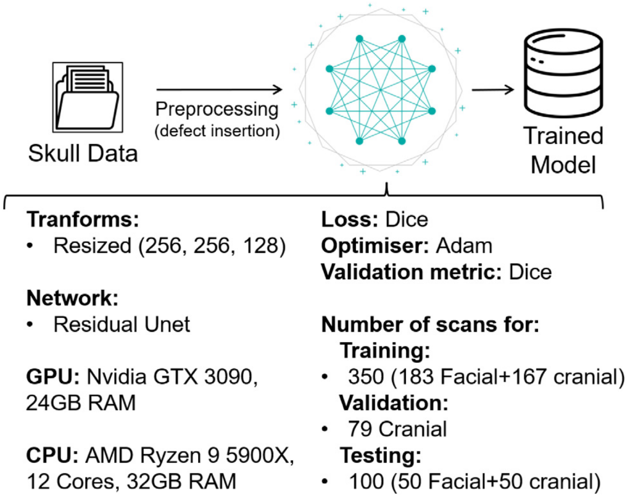

Fig. 2. Main workflow for the creation of the trained skull reconstruction model for MONAI, starting with the preprocessing (defect insertion) of the skull dataset.

To increase the training speed, the MONAI framework provides multithread processing capability to perform data loading and resizing during the training process. It also optimizes resource usage by using the GPU for parallel processing, increasing the training speed compared to normal CPU processing. It is important to note that, it is necessary to have the network and data tensors in the same device, in order to avoid data transfer between different devices.

3. Experiments and results

3.1. Experiments

We trained the autoencoder specified in Section 2.3 using the MUG500+ and SkullFix dataset, respectively, following the model training methods described in Section 3(D). On the MUG500+ dataset, the network is trained for reconstructing both cranial and facial defects, since both defect types are included in its training set. On the SkullFix dataset, the network is trained only to reconstruct facial defects. To test the hypothesis that existing MONAI tutorials can be easily adapted to other applications, we directly used the codes provided in a tutorial on 3D spleen segmentation

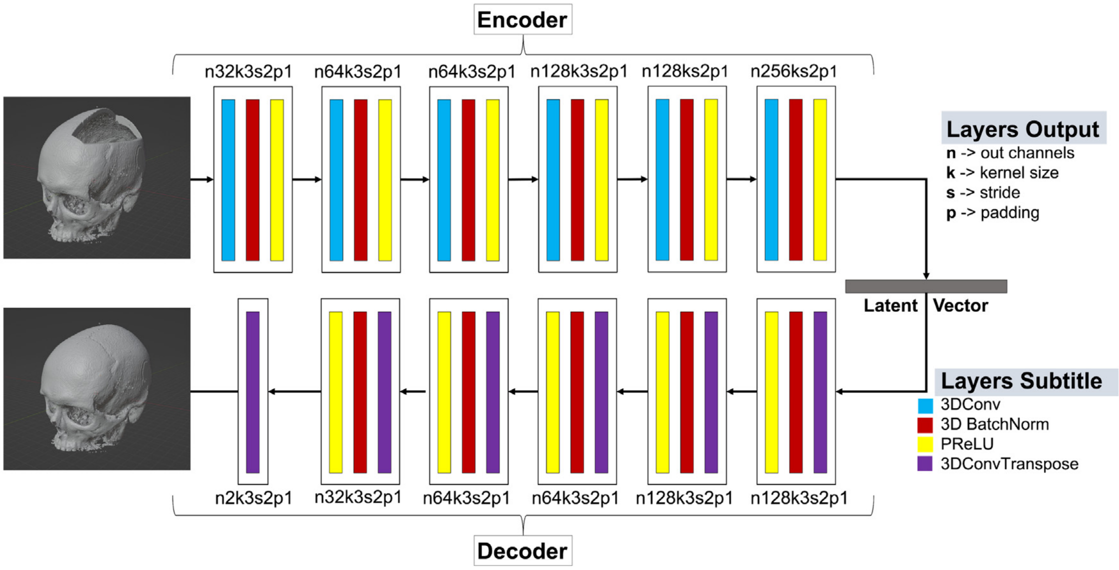

Fig. 3. Structure of the autoencoder used for this contribution. The input of the autoencoder is a skull with a synthetic cranial (or facial) defect and the ground truth is the corresponding complete skull.

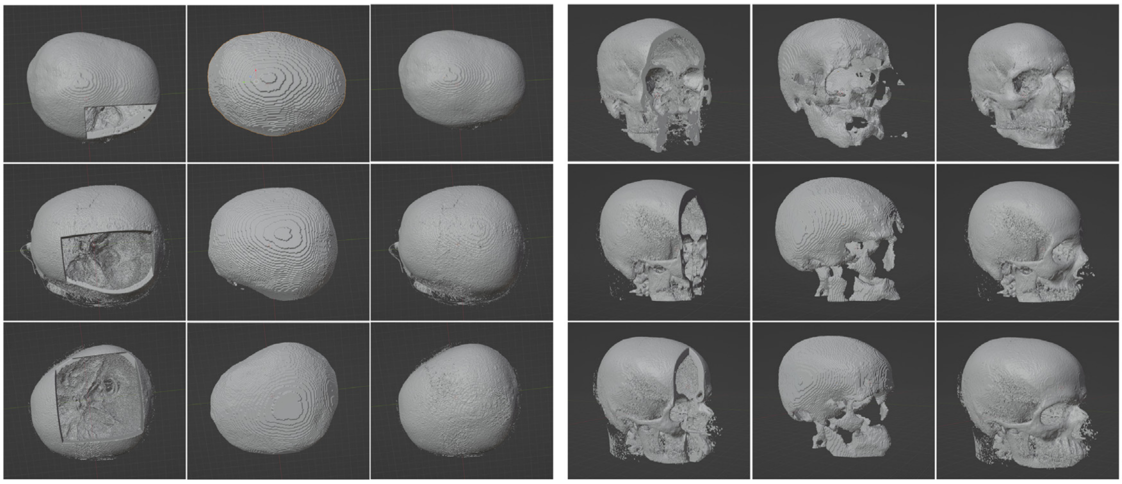

Fig. 4. Reconstruction of the cranial (left) and facial (right) defects on the MUG500+ dataset, using the trained model. The first to third column show the input, the prediction and the ground truth, respectively.

from MONAI. In this experiment, we only made minor changes to the tutorial codes, including the number of layers in the encoder and decoder and the number of output channels in each layer. The paper aims at conducting a feasibility test on MONAI's open-sourcing and easy-integration feature, instead of competing with state-of-the-art methods on skull reconstruction. Therefore, we provide only a qualitative evaluation of the results, as will be discussed in the following section, by inspecting in 3D the quality of the reconstructed skulls.

3.2. Results

Fig. 4 shows the reconstruction of cranial and facial defects for the MUG500+ dataset, using the trained model. We can see that the cranial reconstruction is satisfactory, while the network failed to recover the subtle and complex facial structures. Besides the learning capacity of the network, we attribute the unsatisfactory facial reconstruction performance largely to the MUG500+ dataset itself, as the MUG500+ skulls have obvious artifacts (e.g., spine, catheter extruded from the patients' mouth, etc, as can be seen from the last column of Fig. 4.) that can potentially distract the network from learning the skull geometries. The removal of these artifacts in a preprocessing procedure is non-trivial, since they are closely connected with the skull, and therefore difficult to be separated from the area of interest (e.g., facial bones). One option to better utilize the dataset for cranial implant design is to crop (axially) and discard the entire facial area of the skulls as in [15].

On the contrary, the SkullFix dataset contains mostly artifacts-free skulls, and is more suitable for the facial reconstruction task than MUG500+. The last column in Fig. 5 shows skulls with facial defects. We can see that part or the entire facial structures are missing. The first to third column in Fig. 5 show the facial reconstruction results obtained using the trained network. Note that the initial reconstruction and the input are misaligned (first column, Fig. 5), and therefore the missing facial bones cannot be obtained via a subtraction procedure. We address this by registering the reconstructed completed skull with the input defective skull using a similarity transformation. The second column in Fig. 5 shows the geometry alignment results. Note that the registration is unsuccessful for some cases. The third column in Fig. 5 shows the facial reconstruction in 3D. We can see that, compared

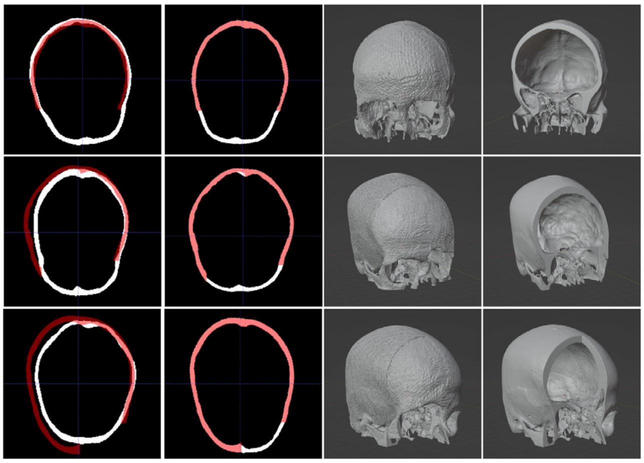

Fig. 5. Facial reconstruction results on the SkullFix dataset. The first to the last column shows the axial view of the reconstruction (shown in white) and input (shown in red) before and after alignment, the reconstructed face and input in 3D, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with the results on MUG500+, the network can learn to restore most of the missing facial structures on the artifacts-free dataset more effectively. Note that the registration step could be avoided by properly aligning the geometry information in the NIfTI output from the model with that of the original NRRD test samples. A 3D viewer for interactive inspection of the skull models can be accessed at https://proj-page.github.io/softwarex_monai.html.

4. Discussion

Deep learning allows to automate tasks that could be done before only manually, or not at all [16,17]. This also enabled new possibilities in the automatic processing of medical images. As demonstrated in the cranial implant design challenges i.e. AutoImplant I at MICCAI 2020 (https://autoimplant.grand-challenge.org/) and AutoImplant II at MICCAI 2021 (https://autoimplant2021.grand-challenge.org/), deep learning-based approaches have shown promising performance in reconstructing skulls where synthetic defects have been inserted. However, deep learning-based approaches still perform unsatisfactorily in reconstructing large and real cranial defects from the clinical routine. In fact, we could recently show that a traditional Statistical Shape Model (SSM) [18,19] can outperform deep learning-based approaches with a fraction of training cases on clinical defective skulls [20]. There are two main reasons for these findings: (1) Synthetic defects only partly resemble real cranial defects, which can be much more complex. In particular, the border area of the defects are more frayed. This, however, can be addressed by creating more realistic-looking synthetic defects. (2) Deep learning-based approaches are data-driven. In general, they work better when more data are involved during the training phase. For the AutoImplant challenges where the SkullFix dataset was used, the participants were provided with only 100 skull pairs for training and we are not aware of participants that used external datasets. The SkullFix dataset is still small for training deep models and a larger skull dataset is desired. A first indication can already be seen from the winning solutions of both AutoImplant challenges [21,22], which drastically expanded the training set through data augmentation. Thus, we assume that our new collection of the MUG500+ skull dataset collection, which was released after the AutoImplant challenges [3,23], will be an impetus for future advancements of learning-based skull reconstruction methods. However, as discussed in Section 3.2, the MUG500+ dataset does not come without its limitations. The facial artifacts that cannot be removed easily pose major challenges for deep learning models to learn the facial geometries effectively. Nevertheless, since this is the first time the MUG500+ dataset is used for skull reconstruction after its release, the data preprocessing strategies introduced in this paper could server as a sound starting-point for other researchers working on the problem.

In addition, MONAI allows us to provide and share an easily accessible pre-trained model on the dataset for the community. This can help to disseminate and address remaining challenges in patient-specific and fully-automatic craniofacial implant design. An example is the usage of automatically designed implants in cranioplasty procedures without major modifications [24]. Another aspect that has to be addressed by the research community is the implant thickness, which should be thinner than the skull bone. Furthermore, because implants design regulations and techniques can vary among different clinical institutions and countries [25,26], federated learning could be used to train a model using datasets from various sources.

5. Conclusion

In this contribution, we presented a pre-trained autoencoder for automatic skull reconstruction as a MONAI research contribution. We showed that existing open-source MONAI tutorials (e.g., '3D spleen segmentation') can be easily adapted to new applications, such as cranial and facial reconstruction. Reasonable results can be obtained without major modification to the sources. The codes and pre-trained models can also be conveniently shared on the MONAI GitHub platform for other MONAI users interested in this topic. The data sets for the pre-training originate from the MUG500+ dataset [2]. All skulls are complete (healthy) with no holes or fractures. Hence, the skulls can be used, for example, by injecting synthetic holes into the healthy skulls and steering an algorithm to learn the task of skull completion [27]. A pure end-user and browser-based solution can be tried out in the online framework StudierFenster (www.studierfenster.at [28]) within the 3D Skull Reconstruction module [29].

Future work sees the pre-training on more data, especially from different clinical institutes that cover a wider variety of CT scanners, scanning protocols, resolutions, etc. Furthermore, a federated learning approach to train algorithms across multiple decentralized edge devices or servers holding local data samples, is desirable, so that researchers can incorporate their own datasets in the pre-trained model, without sharing the dataset. Finally, A model capable of processing multimodal inputs (texts, images, etc.) [30], similar to a real physician, is more desirable for the skull reconstruction task, compared to training on skull images alone. Refer to [31] for another open-source MONAI-based skull reconstruction project about latent space disentanglement [32].

Declaration of competing interest

The authors declare that there are no known competing interests in this submission.

Data availability

Data will be made available on request

Acknowledgment

We acknowledge CAMed (COMET K-Project 871132), enFaced (FWF KLI 678), enFaced 2.0 (FWF KLI 1044) and KITE (Plattform für KI-Translation Essen) from the REACT-EU initiative (EFRE-0801977, https://kite.ikim.nrw/).

References

[1] Cardoso MJ, Li W, Brown R, Ma N, Kerfoot E, Wang Y, et al. MONAI: An open-source framework for deep learning in healthcare. 2022, URL https://arxiv.org/abs/2211.02701.

[2] Li J, Krall M, Trummer F, Memon AR, Pepe A, Gsaxner C, et al. MUG500+: Database of 500 high-resolution healthy human skulls and 29 craniotomy skulls and implants. Data Brief 2021;39:107524.

[3] Kodym O, Li J, Pepe A, Gsaxner C, Chilamkurthy S, Egger J, et al. SkullBreak/SkullFix–Dataset for automatic cranial implant design and a benchmark for volumetric shape learning tasks. Data Brief 2021;35:106902.

[4] Godoy IRB, Silva RP, Rodrigues TC, Skaf AY, de Castro Pochini A, Yamada AF. Automatic MRI segmentation of pectoralis major muscle using deep learning. Sci Rep 2022;12(1):1–9.

[5] Belue MJ, Harmon SA, Patel K, Daryanani A, Yilmaz EC, Pinto PA, et al. Development of a 3D CNN-based AI model for automated segmentation of the prostatic urethra. Acad Radiol 2022.

[6] Shapey J, Kujawa A, Dorent R, Wang G, Dimitriadis A, Grishchuk D, et al. Segmentation of vestibular schwannoma from MRI, An open annotated dataset and baseline algorithm. Sci Data 2021;8(1):1–6.

[7] He Y, Yang D, Roth H, Zhao C, Xu D. DiNTS: Differentiable neural network topology search for 3D medical image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, p. 5841–50.

[8] Yu Q, Yang D, Roth H, Bai Y, Zhang Y, Yuille AL, et al. C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, p. 4126–35.

[9] Hatamizadeh A, Yang D, Roth H, Xu D. Unetr: Transformers for 3d medical image segmentation. 2021, arXiv preprint arXiv:2103.10504.

[10] Wang G, Liu X, Li C, Xu Z, Ruan J, Zhu H, et al. A noise-robust framework for automatic segmentation of COVID-19 pneumonia lesions from CT images. IEEE Trans Med Imaging 2020;39(8):2653–63.

[11] Zhu W, Zhao C, Li W, Roth H, Xu Z, Xu D. Lamp: Large deep nets with automated model parallelism for image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer; 2020, p. 374–84.

[12] Brett M, Markiewicz CJ, Hanke M, Côté M-A, Cipollini B, McCarthy P, et al. nipy/nibabel: 3.2.1. 2020, http://dx.doi.org/10.5281/zenodo.4295521.

[13] Li J, Krall M, Trummer F, Memon AR, Pepe A, Gsaxner C, et al. Mug500+ repository. 2022, https://figshare.com/articles/dataset/MUG500_Repository/9616319.

[14] Schroeder W, Martin K, Lorensen B. The visualization toolkit an object-oriented approach to 3D graphics, kitware. 2, 2003, Inc., Albany, NY, Www. Vtk. Org.

[15] Li J, von Campe G, Pepe A, Gsaxner C, Wang E, Chen X, et al. Automatic skull defect restoration and cranial implant generation for cranioplasty. Med Image Anal 2021;102171.

[16] Egger J, Pepe A, Gsaxner C, Jin Y, Li J, Kern R. Deep learning—A first meta-survey of selected reviews across scientific disciplines, their commonalities, challenges and research impact. PeerJ Comput Sci 2021;7:e773.

[17] Egger J, Gsaxner C, Pepe A, Pomykala KL, Jonske F, Kurz M, et al. Medical deep learning—A systematic meta-review. Comput Methods Programs Biomed 2022;221:106874.

[18] Cootes TF, Taylor CJ. Active shape models—'smart snakes'. In: BMVC92. Springer; 1992, p. 266–75.

[19] Heimann T, Meinzer H-P. Statistical shape models for 3D medical image segmentation: a review. Med Image Anal 2009;13(4):543–63.

[20] Li J, Ellis DG, Pepe A, Gsaxner C, Aizenberg MR, Kleesiek J, et al. Back to the roots: Reconstructing large and complex cranial defects using an image-based statistical shape model. 2022, arXiv preprint arXiv:2204.05703.

[21] Ellis DG, Aizenberg MR. Deep learning using augmentation via registration: 1st place solution to the AutoImplant 2020 challenge. In: Cranial implant design challenge. Springer; 2020, p. 47–55.

[22] Wodzinski M, Daniol M, Hemmerling D. Improving the automatic cranial implant design in cranioplasty by linking different datasets. In: Cranial implant design challenge. Springer; 2021, p. 29–44.

[23] Li J, Gsaxner C, Pepe A, Morais A, Alves V, von Campe G, et al. Synthetic skull bone defects for automatic patient-specific craniofacial implant design. Sci Data 2021;8(1):1–8.

[24] Ellis DG, Alvarez CM, Aizenberg MR. Qualitative criteria for feasible cranial implant designs. In: Cranial implant design challenge. Springer; 2021, p. 8–18.

[25] Campe Gv, Pistracher K. Patient specific implants (PSI). In: Cranial implant design challenge. Springer; 2020, p. 1–9.

[26] Rauschenbach L, Rieß C, Sure U, Wrede KH. Personalized calvarial reconstruction in neurosurgery. In: Cranial implant design challenge. Springer; 2021, p. 1–7.

[27] Morais A, Egger J, Alves V. Automated computer-aided design of cranial implants using a deep volumetric convolutional denoising autoencoder. In: World conference on information systems and technologies. Springer; 2019, p. 151–60.

[28] Egger J, Wild D, Weber M, Bedoya CAR, Karner F, Prutsch A, et al. Studierfenster: An open science cloud-based medical imaging analysis platform. J Digit Imag 2022;1–16.

[29] Li J, Pepe A, Gsaxner C, Egger J. An online platform for automatic skull defect restoration and cranial implant design. In: Medical imaging 2021: Image-guided procedures, robotic interventions, and modeling, Vol. 11598. International Society for Optics and Photonics; 2021, p. 115981Q.

[30] Heiliger L, Sekuboyina A, Menze B, Egger J, Kleesiek J. Beyond medical imaging-A review of multimodal deep learning in radiology. 2022.

[31] Li J, Fragemann J, Ahmadi S-A, Kleesiek J, Egger J. Training β-VAE by aggregating a learned Gaussian posterior with a decoupled decoder. 2022, arXiv preprint arXiv:2209.14783.

[32] Fragemann J, Ardizzone L, Egger J, Kleesiek J. Review of disentanglement approaches for medical applications–towards solving the Gordian knot of generative models in healthcare. 2022, arXiv preprint arXiv:2203.11132.