



Studierfenster: an Open Science Cloud-Based Medical Imaging Analysis Platform

Jan Egger^{1,2,3} · Daniel Wild^{1,2} · Maximilian Weber^{1,2} · Christopher A. Ramirez Bedoya^{1,2} · Florian Karner^{1,2} · Alexander Prutsch^{1,2} · Michael Schmied^{1,2} · Christina Dionysio^{1,2} · Dominik Krobath^{1,2} · Yuan Jin^{1,2,4} · Christina Gsaxner^{1,2} · Jianning Li^{1,2,3} · Antonio Pepe^{1,2}

Received: 10 August 2021 / Revised: 14 December 2021 / Accepted: 16 December 2021 / Published online: 21 January 2022
© The Author(s) under exclusive licence to Society for Imaging Informatics in Medicine 2022

Abstract

Imaging modalities such as computed tomography (CT) and magnetic resonance imaging (MRI) are widely used in diagnostics, clinical studies, and treatment planning. Automatic algorithms for image analysis have thus become an invaluable tool in medicine. Examples of this are two- and three-dimensional visualizations, image segmentation, and the registration of all anatomical structure and pathology types. In this context, we introduce Studierfenster (www.studierfenster.at): a free, non-commercial open science client-server framework for (bio-)medical image analysis. Studierfenster offers a wide range of capabilities, including the visualization of medical data (CT, MRI, etc.) in two-dimensional (2D) and three-dimensional (3D) space in common web browsers, such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge. Other functionalities are the calculation of medical metrics (dice score and Hausdorff distance), manual slice-by-slice outlining of structures in medical images, manual placing of (anatomical) landmarks in medical imaging data, visualization of medical data in virtual reality (VR), and a facial reconstruction and registration of medical data for augmented reality (AR). More sophisticated features include the automatic cranial implant design with a convolutional neural network (CNN), the inpainting of aortic dissections with a generative adversarial network, and a CNN for automatic aortic landmark detection in CT angiography images. A user study with medical and non-medical experts in medical image analysis was performed, to evaluate the usability and the manual functionalities of Studierfenster. When participants were asked about their overall impression of Studierfenster in an ISO standard (ISO-Norm) questionnaire, a mean of 6.3 out of 7.0 possible points were achieved. The evaluation also provided insights into the results achievable with Studierfenster in practice, by comparing these with two ground truth segmentations performed by a physician of the Medical University of Graz in Austria. In this contribution, we presented an online environment for (bio-)medical image analysis. In doing so, we established a client-server-based architecture, which is able to process medical data, especially 3D volumes. Our online environment is not limited to medical applications for humans. Rather, its underlying concept could be interesting for researchers from other fields, in applying the already existing functionalities or future additional implementations of further image processing applications. An example could be the processing of medical acquisitions like CT or MRI from animals [Clinical Pharmacology & Therapeutics, 84(4):448–456, 68], which get more and more common, as veterinary clinics and centers get more and more equipped with such imaging devices. Furthermore, applications in entirely non-medical research in which images/volumes need to be processed are also thinkable, such as those in optical measuring techniques, astronomy, or archaeology.

Keywords Medical image analysis · Cloud · Client/server · Python · ITK · VTK · Deep learning · CNN · GAN · Virtual reality · Augmented reality · Whitepaper

Introduction

In the past few decades, image-based analysis of radiological datasets has gone through a remarkable period of rapid technological innovation. Since imaging modalities, such as computed tomography (CT) and magnetic resonance imaging

✉ Jan Egger
egger@tugraz.at

Extended author information available on the last page of the article

(MRI), are widely used in diagnostics, clinical studies, and treatment planning, automatic algorithms for (bio-)medical image processing and analysis have become an invaluable tool in medicine.

Computers today assist medical decisions in all treatment phases throughout diagnosis, monitoring, therapy planning, execution, and follow-up examinations [1]. Examples are intra-operative navigation in brain tumor surgery [2], radiation therapy planning for cervical cancer [3], or skin tumor segmentation [4]. Segmentation, for example, is typically the first step in a (bio-)medical image analysis pipeline. An incorrect segmentation thus affects any subsequent step. However, automatic medical image segmentation is known to be one of the most complex problems in image analysis and is still an object of active research. Zhang already estimated in 2006 that there are over 4000 image segmentation algorithms [5] and this was well before the advent of the deep learning “era” [6]. Nevertheless, the majority of such algorithms are only available locally and to the research groups that developed them, and an own usage would need both a reimplement and new training data [7]. As a result several medical image processing platforms such as MeVisLab (<https://www.mevislab.de/>) [8], (3D) Slicer (<https://www.slicer.org/>) [9], MITK (www.mitk.org/ and www.radiomics.net.cn/) [10, 11], OsiriX (<https://www.osirix-viewer.com/>) [12], ITK-SNAP (<http://www.itksnap.org/>) [13], RadBuilder [14], XIP-Builder [15], MedAlyVis [16], IBIS (<http://ibisneuronav.org/>) [17], and MeDaS [18] have been established during recent years or even decades, offering common algorithms to the entire community. In doing so, they mostly provide graphical user interfaces to the common and open-source software libraries Insight Segmentation and Registration Toolkit (ITK; <https://itk.org/>) and the Visualization Toolkit (VTK; <https://www.vtk.org/>). Unfortunately, this implicitly means that if a bug is discovered or new library versions are available, the user needs to download and install a new version of the platform. In addition, computer science is currently an incredibly fast evolving field, resulting in recurring updates and new versions, sometimes on a daily basis. To handle this, Slicer provides a so-called nightly build to provide the newest features and bug fixes. The semi-commercial platform MeVisLab is in the meantime now offering new installers that are already much larger than one gigabyte and a problem all platforms have in common is that a user must download a newer version, uninstall the “old” version, and install the newer version from time to time.

An online, cloud-based environment for medical applications, by contrast, would make this process obsolete for end users, as new versions would be deployed and executed on a remote server. In addition, users would not need to worry about the operating system, whereas OsiriX is only offered for macOS/iOS (Apple Inc.). Furthermore, new

developments and algorithms from research groups around the world can also be integrated into this cloud application and would thus be instantly available for use, with rapid feedback and refinement. This would result in an elegant solution for integrating worldwide research findings instantly in a single environment, which is not possible for the existing desktop solutions; or where there is anything whatsoever in this direction, then only as suboptimal extension plug-ins. Such an online environment can also offer an uncomplicated usage of recently arising, portable augmented reality (AR) [19, 20] and virtual reality (VR) devices [21], by accessing the online environment from a VR/AR-ready web browser, such as Mozilla, Firefox, or Google Chrome, thus also removing the need for downloading, installing, and configuring a comprehensive platform package. In summary, an online environment would make the use of AR and VR devices more widespread in the medical domain, especially for medical teaching, training, and web conferencing. The first steps in this direction have already been taken, for example, with an online platform for interactive feedback in biomedical machine learning (<https://www.gradiohub.com>, currently not reachable anymore, last accessed from the authors around July 2021) [22] and Biomedisa, an open-source online platform for biomedical image segmentation (<https://biomedisa.de/>) [23].

Another topic that is currently being actively researched — also for medical applications — is (medical) deep learning [24–26]. For a reliable deep neural network, a massive quantity of training data is needed. This data can be collected globally over a cloud environment and thus be used to build immense training datasets. A single medical dataset can easily reach a few dozen megabytes of storage size, which is not an issue for a cable network connection, but also wireless internet is now able to process these massive volumes, especially with the recently introduced 5G networks.

Material and Methods

In this contribution, we introduce Studierfenster or StudierFenster (www.studierfenster.at), which is a free, non-commercial open science client-server framework for (bio-) medical image analysis. Studierfenster offers a wide range of capabilities, including the visualization of medical data in common web browsers, such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge. Other functionalities are the calculation of common medical scores, manual slice-by-slice outlining of structures in medical images, manual placing of landmarks in medical imaging data, visualization of medical data in VR, and a facial reconstruction and registration of medical data for AR. More sophisticated features include automatic cranial implant design with a convolutional neural network (CNN), the inpainting of aortic

dissections with a generative adversarial network (GAN), and a CNN for automatic aortic landmark detection in CT angiography (CTA) images. The following sections will describe the datasets formats, the overall platform architecture, and the single module of Studierfenster in detail.

Datasets and Preprocessing

Currently, our online platform mainly supports NRRD (nearly raw raster data) image files. These files consist of only the image values (voxel gray values) plus minimal header information and do not include any further information such as the name of the patient or the medical institution where the scan has been performed, which is typically stored in the DICOM (Digital Imaging and Communications in Medicine; <http://dicom.nema.org>) tags. In addition, NRRD files are more convenient to handle, given that they store the whole 3D volume in one single file, in contrast to DICOM, where every scanned slice of the volume is usually stored in a separate file. In summary, NRRD is a file format for the representation and processing of N-dimensional raster data. It is intended to support scientific visualization and (medical) image processing applications [27].

The conversion of the original DICOM files into NRRD files can be easily achieved with tools like MeVisLab, MITK, or Slicer. In general, there are two types of NRRD files: ASCII and binary, which reduces the online traffic, and our web platform only works with compressed binary files. Studierfenster, however, also offers a module to convert non-compressed NRRD files to the corresponding compressed binary versions on its landing page. After conversion, the

compressed NRRD file can be downloaded and used, for example, in the *Medical 3D Viewer* of Studierfenster for 2D and 3D visualization, and further image processing.

Overall Platform Architecture

The overall platform architecture and its communication is shown in Fig. 1. Studierfenster is setup as a distributed application via a client-server model. The client side (front end) has been developed using Hypertext Markup Language (HTML) and JavaScript. The front end also uses the Web Graphics Library (WebGL), a JavaScript Application Programming Interface (API) descending from the Open Graphics Library (OpenGL) ES 2.0 specification, which it still closely resembles. In contrast to OpenGL, WebGL allows the rendering of 2D and 3D graphics in web browsers. This enables the use of graphics features known from stand-alone programs directly in web applications, supported by the processing power of a client-sided graphics processing unit (GPU).

The server side (back end) has been mostly developed in C, C++, and Python (<https://www.python.org/>). It interfaces with common open-source libraries and software tools such as ITK, VTK, the X Toolkit (XTK; <https://github.com/xtk/X>), and Slice:Drop (<https://slicedrop.com/>). The server communication is handled through AJAX requests (https://www.w3schools.com/xml/ajax_intro.asp) [28]. The requests are managed by a Flask server (<https://flask.palletsprojects.com>) that handles the exchange of image data, abstract data structures, like seed point positions or segmentation contours, and takes care of partial progress reporting when algorithms are executed on the server. The Flask server connects to

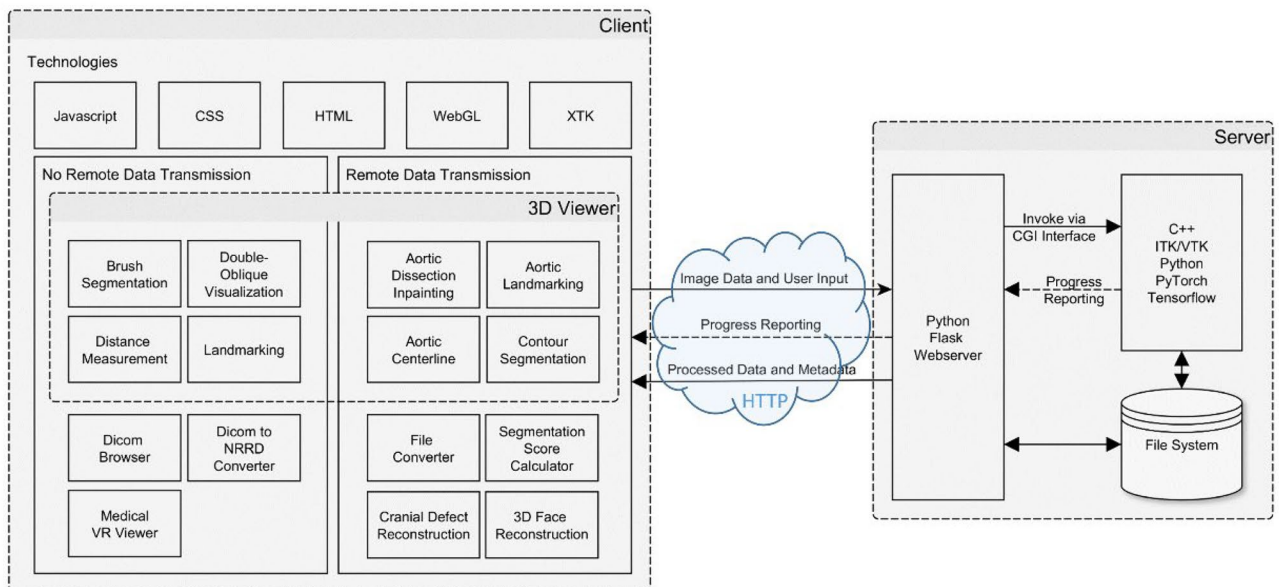


Fig. 1 Overall platform architecture of Studierfenster with its modules and communications

the various algorithms, implemented in C++, ITK, VTK, PyTorch (<https://pytorch.org/>), or TensorFlow (<https://www.tensorflow.org/>), and also communicates with the file system on the server.

For a distributed multi-user development, a Vagrant box (www.vagrantup.com) in combination with a GitLab repository (www.gitlab.com) was set up. The Vagrant box is platform-independent and thus allows contributors to use their preferred and installed operating systems, such as Windows, macOS, or Linux. In addition, users work on local developer branches and *push* their final implementations to the main branch once fully implemented and tested to go online. The online environment, which is currently hosted at the Graz University of Technology in Austria (<http://studierfenster.tugraz.at/>), can already be used to perform tasks like medical visualizations and manual segmentations directly in the web browser. The manual segmentation workflow, for example, consists of dragging and dropping the input file into the browser window and outlining the object under consideration in a slice-by-slice fashion. The final segmentation can then be exported as a file, storing its contours or a binary segmentation mask. The following sections will introduce several single Studierfenster modules in more detail and the current landing page of Studierfenster is shown in Fig. 2.

DICOM Browser and Converter

As stated above, Studierfenster works mainly with an NRRD image. However, we also offer a purely client-sided DICOM browser and converter [29] (Fig. 3). The DICOM browser allows client-sided parsing of a zipped local folder with DICOM files. Subsequently the whole folder can be converted to compressed NRRD files and downloaded as a single .zip file (Fig. 4). These resulting NRRD files contain no patient tags, like name and age, and because the DICOM browser and converter is purely client-sided (in fact, it is loaded completely into the cache of the web browser of the user when accessing the Studierfenster website), the DICOM files and patient tags are never transferred to our server during the conversion process.

Using the DICOM browser of Studierfenster, it is also possible to select specific studies or series exclusively and to convert only these. The DICOM module has been developed in JavaScript and when the Studierfenster website is accessed, it is loaded into the cache of the web browser. This avoids further server communications and ensures that no DICOM data is transferred to the Studierfenster server, also during the conversion process.

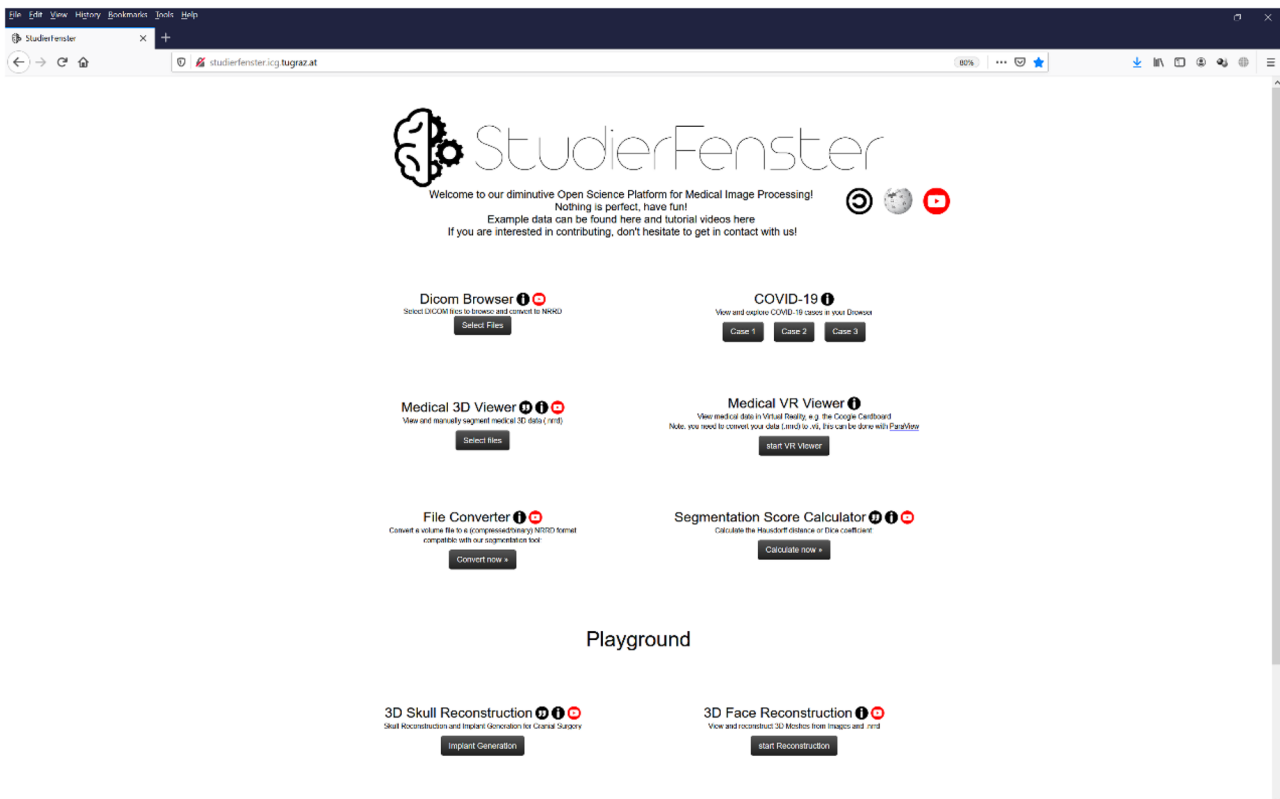


Fig. 2 The current Studierfenster landing page

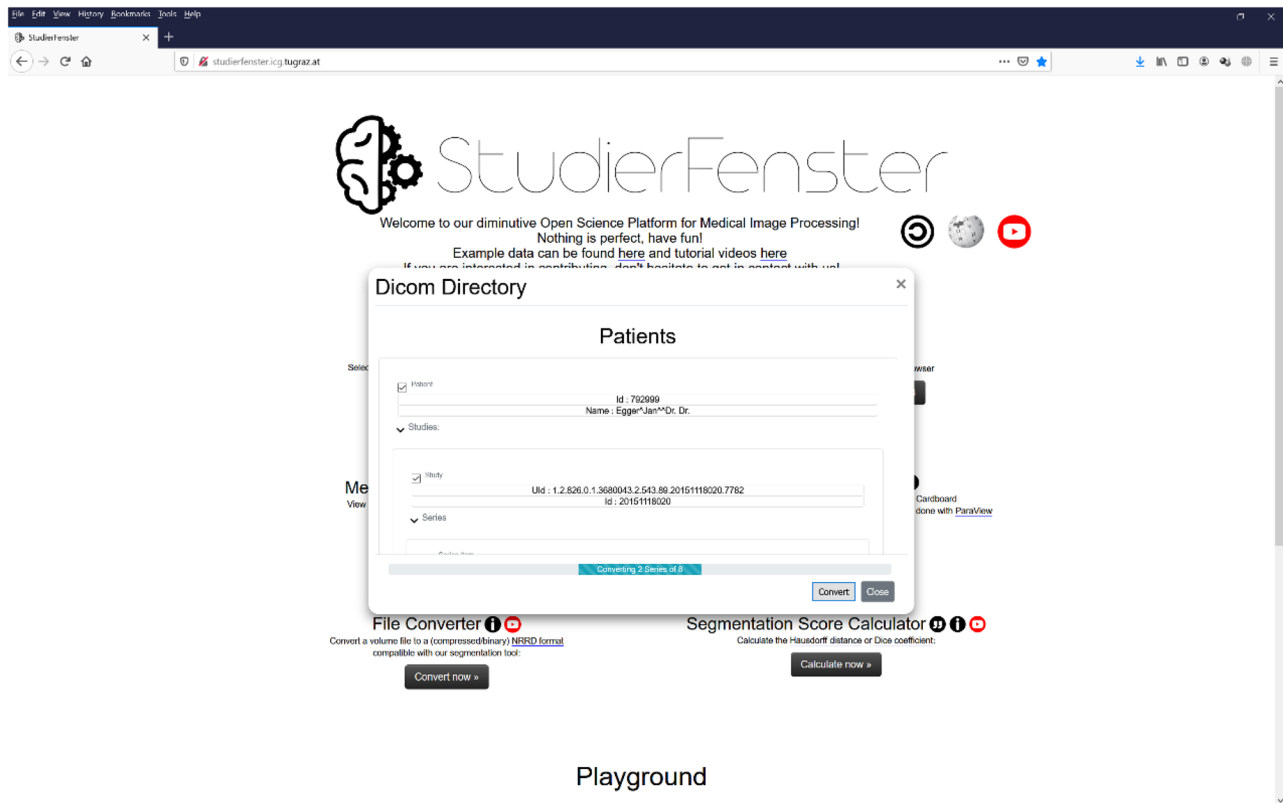


Fig. 3 Client-sided DICOM browser and converter from Studierfenster [30]

2D and 3D Data Visualization

The core of Studierfenster is currently the so-called *Medical 3D Viewer* module. It enables the visualization of 2D and 3D data in a standard web browser without further server communication. It is based on XTK and offers for (medical) volume data the classical 2D views in axial, coronal, and sagittal directions (Fig. 5). In addition, XTK already comes with options such as thresholding and cross-sectional slicing of 3D image data, and volume rendering, which is achieved via WebGL. For Studierfenster these functionalities have been extended and the viewer has been connected to several own client- and server-sided modules, for example, by manual contouring and landmarking capabilities of anatomical structures or pathologies in (bio-)medical datasets or even the fully automatic processing of (bio-)medical datasets.

Manual Annotation

Within the *Medical 3D Viewer*, Studierfenster offers a functionality for client-sided manual annotation of the imaging datasets [33]. On the one hand, this can be the manual outlining (contouring) of anatomical or pathological structures (Fig. 6); on the other hand, this can be the manual placement of landmarks within the image or volume. The main usage

for these functionalities is to generate ground truth annotations within medical datasets. These can be, for example, used afterward to evaluate the automatic annotation results produced by automatic algorithms. The annotations are purely client-sided, which means no further server communications are needed and the medical data does not need to be transferred to the Studierfenster server. This is particularly important for all cases where local legislation does not allow a data upload to third-party servers.

The annotations can be saved locally as CSV files. However, the CSV files can later be re-loaded in Studierfenster to continue with the annotation or for visualization. Because these functionalities work in a standard web browser, they can be used, for example, by physicians to generate ground truths within a very restricted hospital environment, which, in general, does not permit the installation of (external) segmentation tools or software, like Slicer. In addition, Studierfenster offers a brush-based manual segmentation module, through which segmentation results can be saved locally (and re-loaded) as NRRD masks.

Automatic Aortic Landmark Detection

In addition to the manual placement of landmarks in medical datasets, Studierfenster also offers a fully automatic

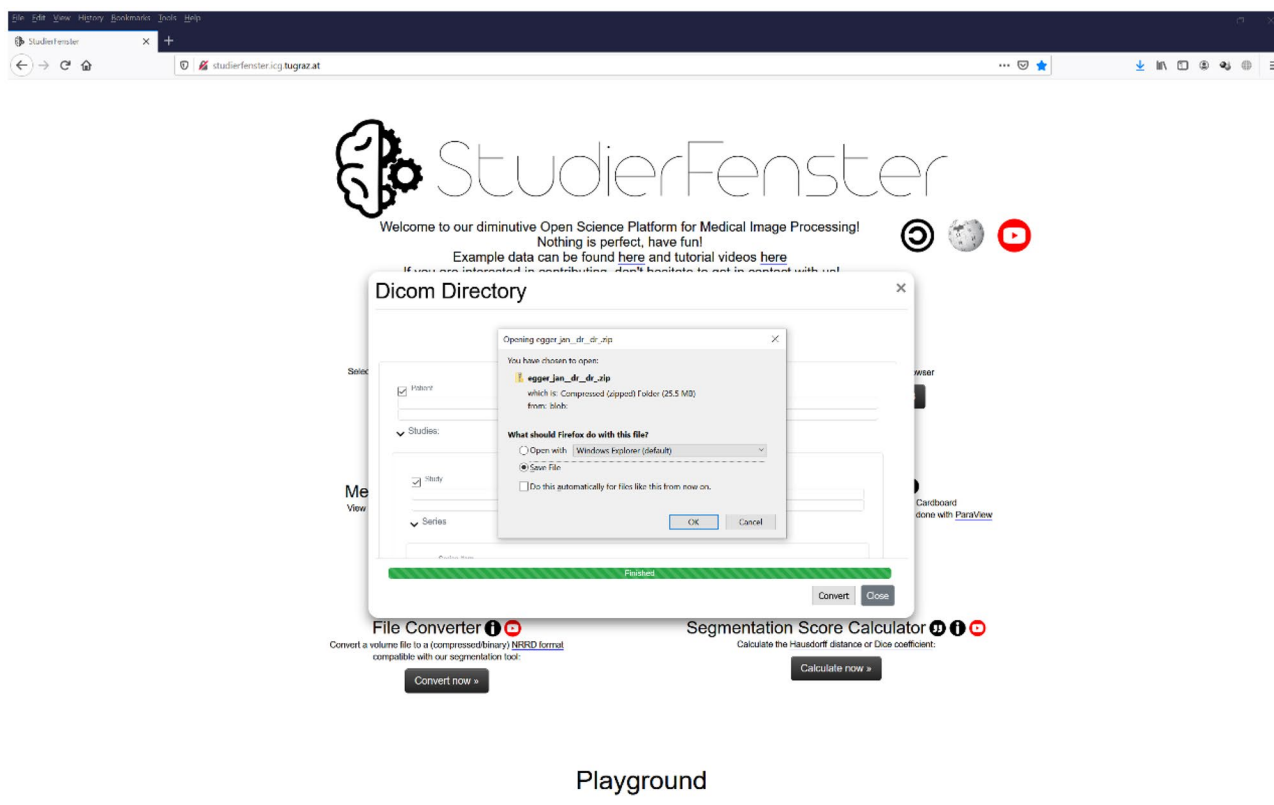


Fig. 4 After the selection and conversion of specific (or all) studies or series to compressed .nrrd (nearly raw raster data) files, these can be downloaded as a single .zip file

detection of aortic landmarks in CTA datasets of the thorax, which we initially developed in the course of the TU Graz Lead Project on Aortic Dissections (<https://www.tugraz.at/projekte/aortic-dissection/home/>) [38]. The landmarks were used in this project as a starting point for the tracking of the aortic centerlines. The automatic landmark detection utilizes a patch-based CNN [39] that runs on the Studierfenster server and that can predict different landmarks simultaneously. In more detail, Studierfenster uses a patch-based iterative network (PIN-algorithm) [40], which combines a classification and a regression problem in one joint network. Thereby, random initial landmarks move towards a “true” landmark by multi-task learning, predicting the magnitude and direction of movement. The functionality is accessible via the *Medical 3D Viewer* of Studierfenster, but needs the transfer of the medical dataset (as anonymized NRRD) to the Studierfenster server for processing.

Aortic Dissection Inpainting

Studierfenster also offers an inpainting functionality for aortic dissections [41] based on a GAN [42] within the *Medical 3D Viewer* [43]. In more detail, it is a semi-supervised virtual regression of aortic dissections, which removes the so-called false lumen and reconstructs an approximation

of the healthy aorta [44]. In doing so, a two-stage GAN is used. The first stage reconstructs the aortic wall and depends on a network that is trained on the edge information of the healthy aortas. The second stage reconstructs the entire dataset including the texture by inferring the image information of the aorta. The inpainting can be performed by a user on specific axial slices of a volume loaded with the 3D viewer in Studierfenster showing an aortic dissection. Afterward, the inpainted volume can be downloaded by the user as NRRD. Corresponding dissected and healthy cases (for the same patient), which in general do not exist in clinical practice, can be used for further analysis of aortic dissections and their causes and disease development.

Centerline Tracking

The centerline functionality of Studierfenster allows the automatic calculation of a vessel’s centerline in a 3D volume [45]. A vessel’s centerline can be a practical method to analyze blood vessels, like the aorta, and, for example, evaluate its shape in a CT or CTA scan. The cloud-based centerline tool of Studierfenster has mainly been developed and tested for the aorta, but can also be tried out on other vascular structures. It has been implemented within the *Medical 3D Viewer* of Studierfenster and requires the user to place two

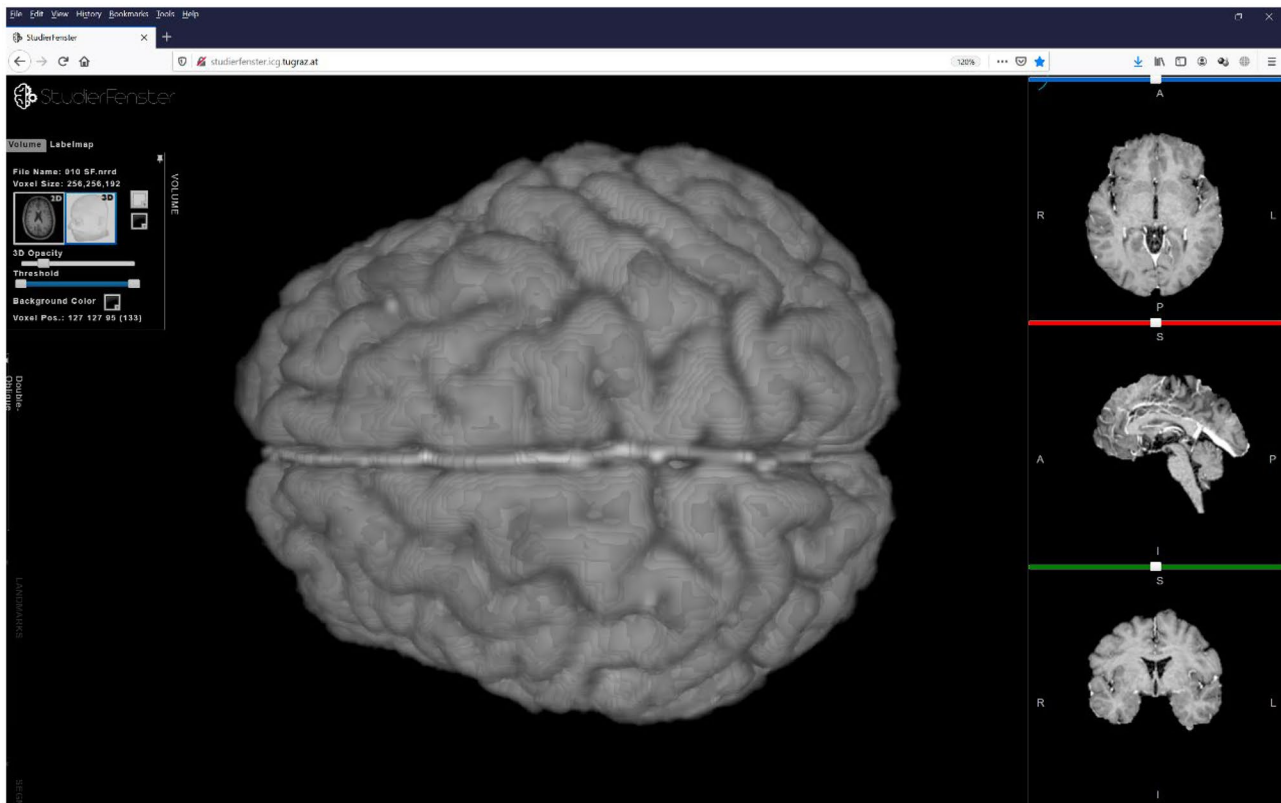


Fig. 5 Medical 3D Viewer of Studierfenster with the classical 2D views in axial, coronal, and sagittal directions (right) and volume rendering (middle) [31, 32]

seed points (as start and end points of the centerline) within the vessel. Following on from this an initial centerline of the vessel, e.g., the aorta, is computed (Fig. 7, left). In a subsequent step, the user has the additional option to further and automatically smooth this initial centerline (Fig. 7, right). Finally, the user can locally download the centerline as a CSV file, but also import an existing centerline into Studierfenster for visualization or further processing.

3D Skull Reconstruction

Studierfenster also offers a module for the automatic reconstruction of skull defects, on which we work currently within the CAMed project (<https://www.medunigraz.at/camed>): a skull dataset of a patient with a cranial defect/hole can be uploaded to the Studierfenster server and the hole will be automatically filled to restore a healthy skull (i.e., skull shape completion) [46]. In addition to this, Studierfenster can also then subtract the defective skull from the restored/completed skull. The result (i.e., the difference) can be downloaded in the STereoLithography (STL) format for further processing and, for example, additive manufacturing/3D printing (Fig. 8). Integrated into clinical practice, this would enable a very rapid, patient-specific in-house cranial implant design, a process which is currently

still outsourced to external products in clinical routine with all the drawbacks that this involves [48]. The fully automatic deep learning-based algorithm has been self-supervised during training by injecting artificial defects in healthy skulls [49, 50]. An in-depth review of algorithms for an automatic cranial implant design can be found in *AutoImplant 2020* summary paper [51] and challenge proceedings [52].

3D Face Reconstruction and Registration

Studierfenster offers also a reconstruction and registration module for medical applications. In more detail, the module enables the reconstruction of a 3D model from a single 2D photo of a person's face [53], which can be automatically registered to the medical head/face scan of this person (Fig. 9). This approach was later used in a non-real-time mobile medical augmented reality application for the facial area [54]. This allows an augmented visualization of anatomical and pathological information in a video see-through fashion on the mobile screen, accurately registered and overlaid with the live patient. In summary, this permits a cost-effective, marker-less, and easy-to-use approach, in contrast to other systems that rely on additional markers, external devices, or depth sensors [55].

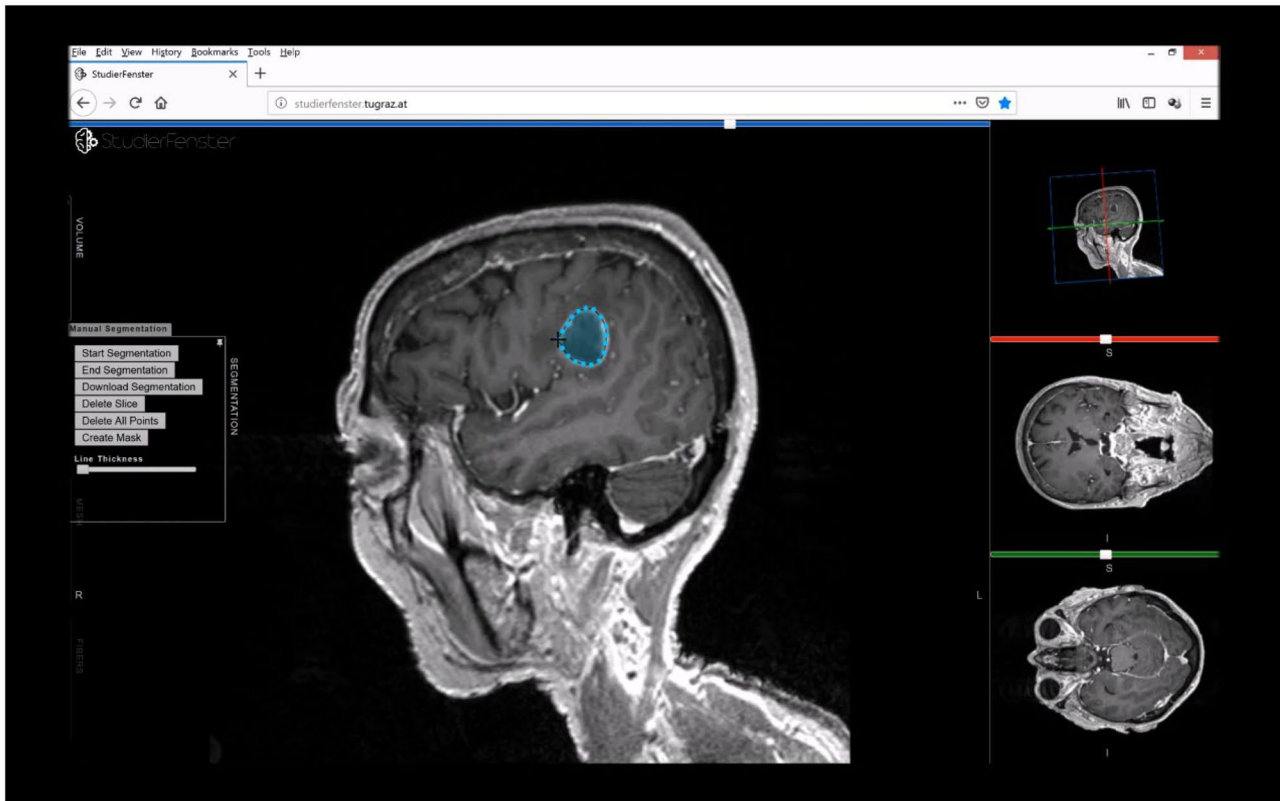


Fig. 6 Manual segmentation of a brain tumor (glioblastoma multiforme (GBM), blue) in a magnetic resonance imaging (MRI) scan of a patient [37]

Medical Virtual Reality Viewer

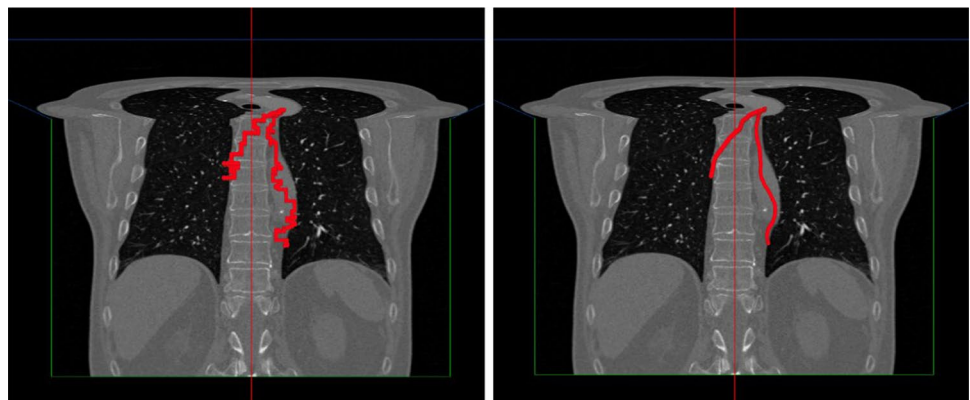
The *VR Viewer* (or *Medical VR Viewer*) module of Studierfenster enables viewing (medical) data in VR with devices like the Google Cardboard or the HTC Vive (via the WebVR App). For viewing the data in VR, it needs to be converted to the VTI (.vti) format, which can be done within ParaView (<https://www.paraview.org>). In contrast to other options, like viewing medical data under Unity (<https://unity.com/>) [56], which needs the installation of Unity or an App itself and dealing with specific plug-ins

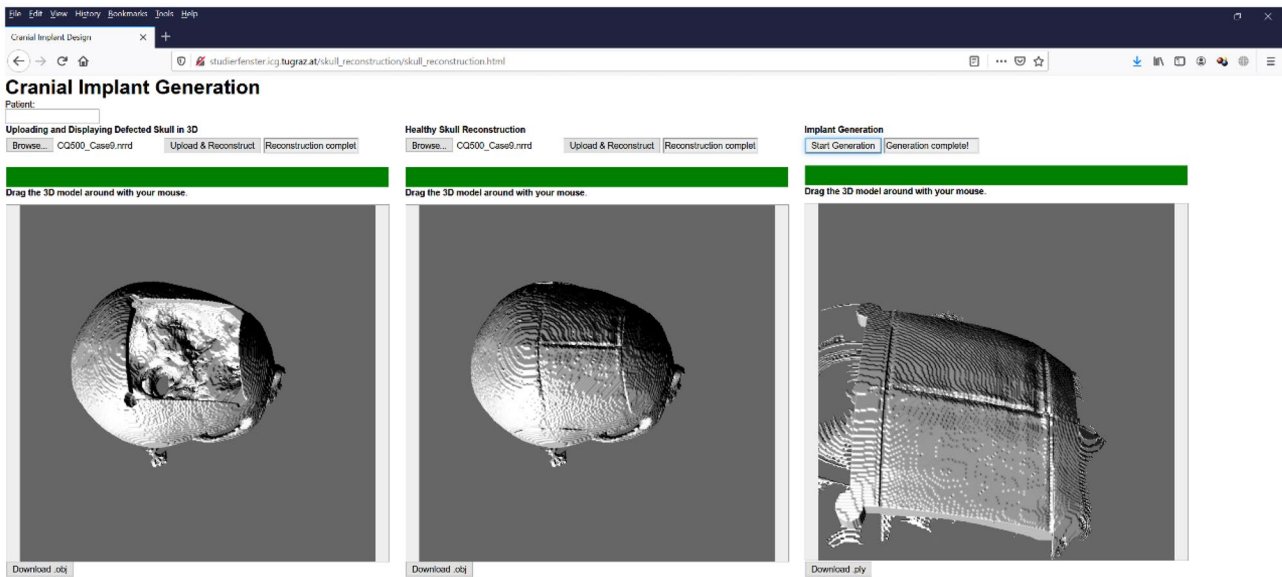
and also a file conversion, the Studierfenster option is quite lightweight. The same applies for other software tools that need to be installed, like MeVisLab [57].

Dice Coefficient and Hausdorff Distance Calculation

Studierfenster also provides a functionality to calculate dice similarity coefficient (DSC) [58] and directed and undirected Hausdorff distance (HD) [59] scores for two uploaded volumes (Fig. 10) [60]. We decided for the DSC and HD, because these two metrics are the most popular and widely

Fig. 7 An initial centerline (left, red) and the corresponding smoothed centerline (right, red) calculated and visualized with Studierfenster [45]





A test case can be found [here](#). Computation times:

1. Window: ca. 40 seconds
2. Window: ca. 2:35 minutes
3. Window: ca. 19 seconds

Fig. 8 Skull reconstruction under Studierfenster: defected skull (left window), reconstructed skull (window in the middle), and subtraction (right window) [47]

used ones in the (bio-)medical community. However, based on our implementation, other scores can easily be added in the future. An overview of medical scores can be found

in the publication of Taha and Hanbury [61]. The Dice coefficient, also known as the Sorensen dice coefficient, is actually the most used metric for validating (bio-)medical

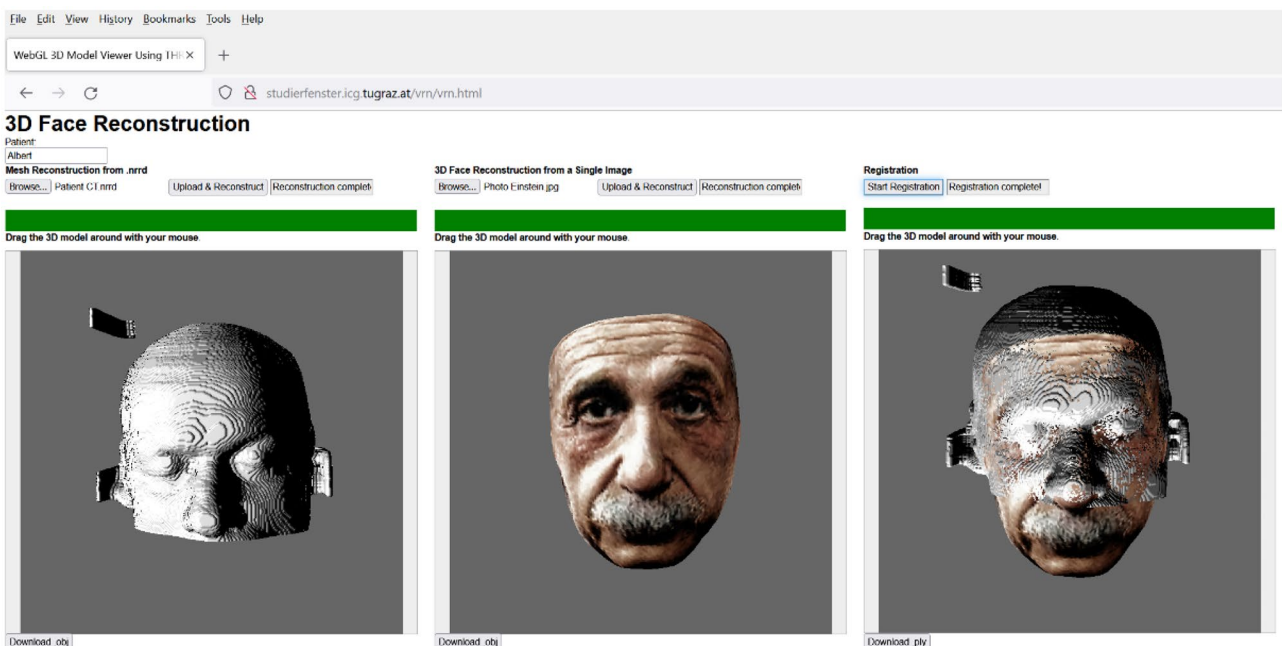


Fig. 9 3D Face Reconstruction and Registration module of Studierfenster: extracted surface of a medical head/face CT scan (left window), reconstructed 3D model from a single photo from a person's face (window in the middle), and registration of both 3D models (right window)

image segmentations. It is an overlap-based metric, and for a ground truth segmentation S_g and a predicted segmentation S_p , the DSC can be calculated as

$$DSC = \frac{2|S_g \cap S_p|}{|S_g| + |S_p|}$$

where $2|S_g \cap S_p|$ is the intersection between ground truth segmentation and predicted segmentation, which corresponds to the true positives (TP). $|S_g|$ and $|S_p|$ denote the total amount of pixels/voxels classified as foreground in the ground truth segmentation and the predicted segmentation, respectively. The DSC takes values between zero and one, whereby one equals a perfect match (100% overlap between S_g and S_p).

The HD, on the other hand, is a spatial distance-based similarity measure, which means that the spatial positions of the pixels/voxels are taken into account, and the HD between point set A and point set B is defined as

$$HD(A, B) = \max(h(A, B), h(B, A))$$

where $h(A, B)$ is the directed HD that describes the maximal distance of the point set A to the closest point in the point set B:

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

where a and b are points of point set A and point set B, respectively, and $\|\dots\|$ is a norm, in example a L2 norm to calculate the Euclidian distance between the two points, e.g.,

$$\|a - b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$$

$h(A, B)$ is the distance between the most distant point of point set A from the closest point of point set B and vice versa for $h(B, A)$. Finally, HD represents the maximum between $h(A, B)$ and $h(B, A)$ [62].

After the calculation of DSCs and HDs for several volumes, Studierfenster offers the options to filter or search for specific values in the calculated metrics, and these filtered metric lists can be exported in different file formats, like CSV, Excel, and PDF (Portable Document Format) (Fig. 11).

Data Availability

The datasets generated during and/or analyzed during the current study are available in the Figshare repository (https://figshare.com/authors/_/3752557).

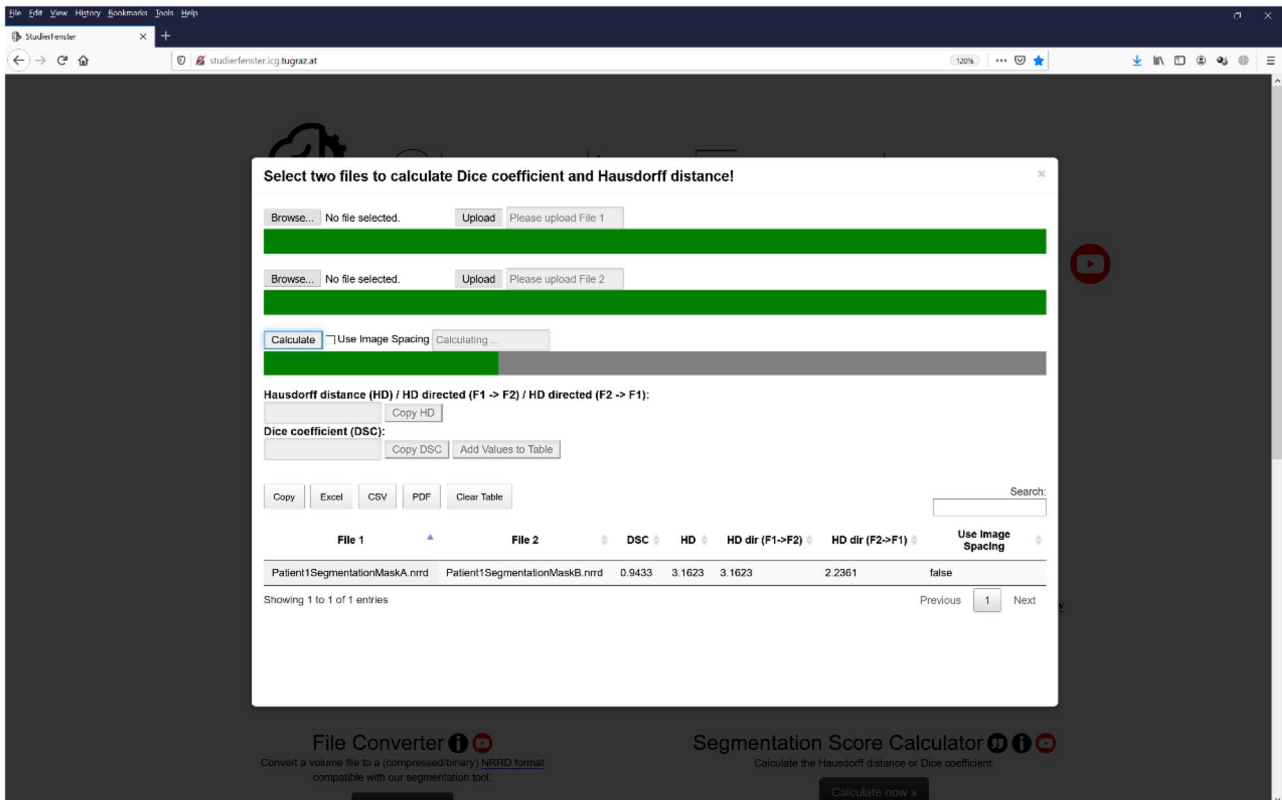


Fig. 10 Studierfenster functionality of calculating dice similarity coefficient (DSC) and directed and undirected Hausdorff distance (HD) scores for two uploaded volumes

File 1	File 2	DSC	HD	HD dir (F1->F2)	HD dir (F2->F1)	Use Image Spacing
segmentation02.mha	segmentation04.mha	0.8548	5.4844	4.2482	5.4844	false
segmentation02.mha	segmentation03.mha	0.8501	4.1231	2.4495	4.1231	false
segmentation05.mha	segmentation02.mha	0.8317	5.2029	2.4527	5.2029	true

Showing 1 to 3 of 3 entries (filtered from 5 total entries)

Previous 1 Next

Fig. 11 After the calculation of DSCs and HDs for several volumes, our tool provides the options filtering for specific values, searching for specific values in the calculated metrics, and exporting filtered metric lists in different file formats, such as CSV, Excel, and PDF

Results

The single modules and functionalities have been evaluated with various medical datasets from different modalities, anatomies, and pathologies. The DICOM browser and converter have been tested with internal and external publicly available DICOM data collections, like [63]. Average computational times have been measured, but also a breakdown of sub-processes in the conversion pipeline for an ablation study has been performed [29]. The resulting and converted NRRD files have been compared to the conversion results of other offline platforms mentioned in the introduction. We currently have two configurations as server hardware, one VPN (virtual private network) machine (Ubuntu) with no GPU access running on <http://studierfenster.tugraz.at/> and a second server with an Intel Core i7-7700K at 4.2 GHz and 64GB DDR4-RAM and GPU access to a GeForce 2070 with 8GB RAM running on <http://studierfenster.icg.tugraz.at/> (www.studierfenster.at currently forwards to the server with the GPU by default).

For an automatic landmark detection, the PIN-algorithm has been evaluated via the localization mean error and standard deviation for 1000 independent predictions of four landmarks: one landmark is located close to the entrance of the heart's left ventricle in the ascending part of the aorta; another one was placed in the center of the aortic arch; the third landmark was on the height of the first landmark, but positioned on the descending branch of the aorta (proximal descending aorta); and the fourth and final landmark was placed near the bottom of the scan in z -direction (distal descending aorta), about six slices from the lower end of the scan. For generating the ground truth, all four landmarks were individually placed by hand for all 75 CTPA (Computed Tomography Pulmonary Angiography) scans (http://www.cad-pe.org/?page_id=14.Online) [66] with the manual landmark annotation functionality of Studierfenster. The network's prediction accuracy was quantitatively evaluated and the best setup achieved an average landmark localization error of 16.04 mm. Thereby, 99.0% of all predictions had a

mean error smaller than 30 mm, while the best prediction during evaluation had a mean error of 3.53 mm over all landmarks [38].

The *Inpainting* module of Studierfenster has been evaluated with different browsers with a specific focus on the code execution on the client side, which took only between 40 and 60 ms. An in-depth ablation study showed that it takes between 3.7 and 3.9 s to fulfill the HTTP request to the Studierfenster server. Further time measurements showed that the execution of the Flask Server code produces only a small impact of 3% of the whole request duration and the share of the data transfer network was also relatively small with 5%. With 92%, executing *EdgeConnect* needed the vast majority of the time to fulfill the HTTP request and an overall inpainting task took around 4 s [43]. Furthermore, we trained our two-stage GAN with 3D patches from 55 non-dissected aortic datasets and evaluated it on 20 more non-dissected datasets, which we plan to add to Studierfenster in the near future [44].

In order to evaluate the *Centerline* module of Studierfenster, we tested the smoothed centerlines computed within Studierfenster against 40 baseline centerlines from the public available CAD-PE Challenge dataset that we used also for evaluating the automatic landmark detection module [45]. The baseline centerlines were extracted by us in a semi-manual process using a customized MeVisLab network. For the comparison between a ground truth and a Studierfenster-generated centerline, we computed the minimum, maximum, and mean distances between them in millimeters, which resulted in the smallest distance of 0.59 mm, an overall maximum distance of 14.18 mm, and a mean distance for all cases of 3.86 mm with a standard deviation of 0.99 mm. The centerline computation times for Studierfenster varied between 18 and 55 s on our current hardware setup, depending on the input file size and the location of the seed points inside the aorta. The centerline smoothing in the second step took between 3 and 5 s and depended on the number of initial centerline points. The initial centerlines were smoothed with a weight parameter of 1, a tolerance value of 0.00001, and a maximum iteration number of 1000.

The Studierfenster *3D Skull Reconstruction* module has been qualitatively and quantitatively evaluated on the publicly available CQ500 dataset (<http://headctstudy.qure.ai/dataset/>) [64]. In this process large, artificial defects have been injected into the (healthy/complete) skulls to create corresponding defective skulls [46]. Following on from this the *Skull Reconstruction* module of Studierfenster has been utilized to reconstruct the defected skulls again, or in other words, to fill the artificial injected holes again with “bone.” The evaluation between the ground truth skulls (original CQ500 skulls) and the reconstructed skulls from Studierfenster yielded a mean Hausdorff distance between 0.5328 and 1.4917 voxels, and a mean RMS (root mean square) between 0.7998 and 2.0154.

The datasets from [65] were used to verify the calculation results of the *Metrics* module (*segmentation score calculator*) achieved with Studierfenster. This dataset collection consists of manual segmentations of mandibles in CT scans from ten patients. Thereby, every mandible has been segmented twice by two facial surgeons and the dice similarity coefficients and Hausdorff distances between two corresponding manual segmentation masks have been calculated with MeVisLab [60]. The resulting scores have been compared with the results from Studierfenster, where the dice similarity coefficients and Hausdorff distances between the same two corresponding manual segmentation masks have also been calculated. As a result, Studierfenster delivered exactly the same metric results as the widely used desktop application MeVisLab.

The *Virtual Reality* module of Studierfenster has been tested with a Google Cardboard, the HTC Vive, and an Oculus Rift, for technical functionality. As datasets, we mainly used cases from public collections, such as [65]. Moreover, we linked three confirmed CT scans of COVID-19 cases into Studierfenster, which can be opened with just one click in the *3D Viewer* by an interested user. After this, it is possible to explore these cases in 2D (in axial, sagittal, and coronal directions) and 3D. The COVID-19 cases come from a public dataset collection and more details can be found under the following website: <http://medicalsegmentation.com/covid19/>.

In order to evaluate the usability and the manual functionalities of Studierfenster, a user study with medical and non-medical experts in medical image analysis was performed. The user study involved ten users in total and participant’s informed consent was obtained. Five of these users were familiar with other current existing software solutions and the other five users had no prior experience in medical image analysis. When asked about their overall impression of Studierfenster in an ISO standard (ISO-Norm) questionnaire, a mean of 6.3 out of 7.0 possible points was achieved [33]. Answers were given on a Likert scale ranging from one to seven, where one is

the worst rating and seven the best. The evaluation also provided insights into the results achievable with the Studierfenster in practice, by comparing them with two ground truth segmentations performed by a physician of the Medical University of Graz in Austria. Figure 12 presents the mean ratings given in the questionnaire and the corresponding standard error as a bar chart for the following questions:

1. The software does not need much training time.
2. The software is well adjusted for the achievement of a satisfying result.
3. The software provides all the necessary functions for achieving the goal.
4. The software is not complicated to use.
5. How satisfied are you with the UI surface?
6. How satisfied are you with the presented result?
7. How satisfied have you been with the time required?
8. What is your overall impression?

The user study consisted of two parts: (1) a short introduction to the tool, where we gave the participants the opportunity to freely explore the tool. This initial training and familiarizing took about 5 min on average. (2) The actual segmentation task, which took the participants between 3 and 15 min to finish. Here, most of the variance comes from the medical group, which took on average of 8 min and 47 s, with participants well distributed between the 3- and 15-min mark. In contrast, all but one participant from the non-medical group took between 3 and 4 min to complete the segmentation task. The one “outlier” took 14 min and 32 s. The main reason for the time difference between the two groups is that participants of the medical group edited and refined their segmentation contours more frequently [33].

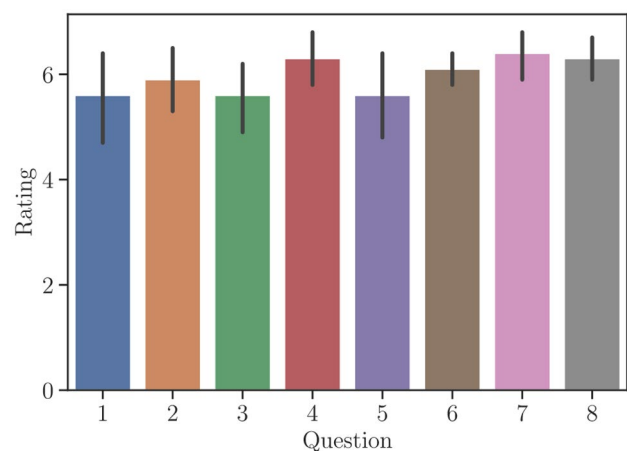
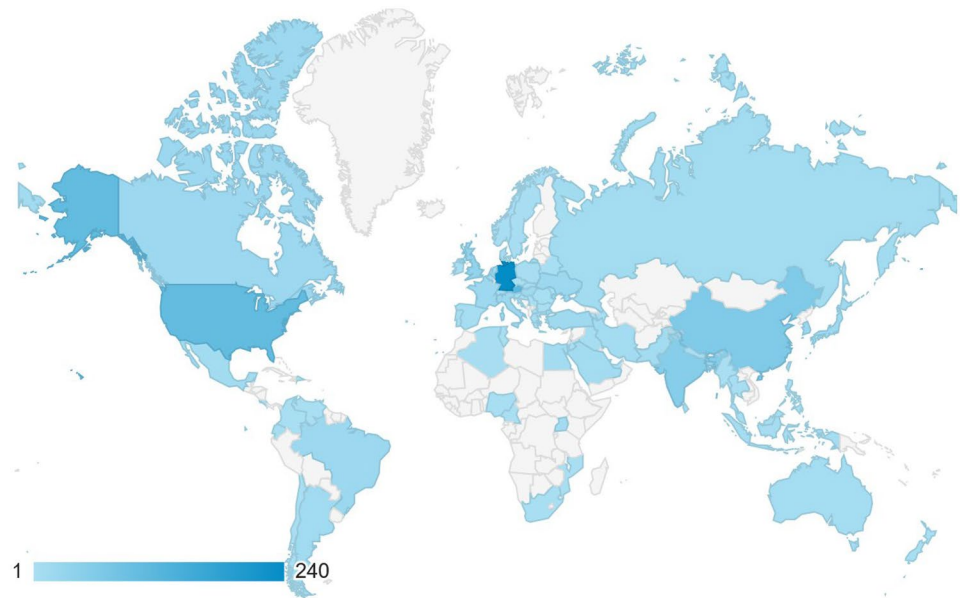


Fig. 12 User study results visualized as a bar chart, presenting the mean of the ratings of all users grouped per question [33]

Fig. 13 Map of the worldwide distribution of the first 1000 users, accessing and interaction with Studierfenster (monitored with Google Analytics)



Conclusions

In this contribution, we presented an online environment for (bio-)medical image analysis. In doing so, we established a client-server-based architecture, which is able to process medical data, especially 3D volumes. For elementary capabilities, like the visualization and annotation of medical imaging data, no server connection and interchange are needed, everything take place in a standard web browser on the client side. For more sophisticated features and algorithms, such as statistical calculations, centerline tracking, cranial implant generation, inpainting, and deep learning tasks, data is transferred and processed via Studierfenster. After this, the processed results, which can be an image, volume, landmarks, or a centerline, are send back to the client for visualization and further processing, and downloading. Our online environment works mainly with data in the anonymized and compressed NRRD format, to reduce the client/server network traffic. By anonymized we mean that no patient DICOM tags, such as name and age, are transferred to our server and an automatic routine ensures that all uploaded data is deleted on a daily basis. The user is still responsible, however, to ensure that only data is processed, which is already publicly available, like [65–68] and that the user complies with all the local regulations, such as an approved institutional review board (IRB), and is allowed to process the (bio-)medical data with our online environment.

Our online environment is not limited to medical applications for humans. Its underlying concept is rather something that could be interesting for researchers from other fields, in applying the already existing functionalities or future additional implementations of further image processing

applications. An example could be the processing of medical acquisitions like CT or MRI from animals [69], which are becoming more and more common, as veterinary clinics and centers are becoming ever more widely equipped with imaging devices of the kind. Furthermore, use in entirely non-medical research in which images/volumes need to be processed is also thinkable, as for example in optical measuring techniques [70], astronomy [71], or archaeology [72].

Further work in the near future envisions the adding of more functionalities to our online environment, for example, in the online design of cranial or facial implants with task-specific workflows [73–75]. In addition, we need to address a software testing approach, for instance, an automated method for regression testing. In parallel, we invite other researchers and research groups to collaborate and join our research playground. Note that there is currently no public API for Studierfenster available, so interested external researchers or developers need to get in contact with us and we will provide them an API or integrate their algorithms.

Finally, we started monitoring the worldwide access and usage of our Studierfenster server with Google Analytics. In this regards, Fig. 13 shows a map of the worldwide distribution of the very first 1000 users, accessing and interaction with Studierfenster. Interestingly, most of the users were from Germany, which may be attributed to the German origin of the chosen name for our framework: Studierfenster. In addition, we noticed that Studierfenster has been used, for example, in the Dice coefficient calculations to compare manual anatomical segmentations of brain tumors [34], listed under softwares and techniques in a book about data analytics [35] and stated in a review about published literature on systems and algorithms for the classification,

identification, and detection of white matter hyperintensities of brain MR images [36]. In the future and when more users are involved, we also plan to monitor more “access details,” such as user access to different Studierfenster transactions/tasks, which will provide more insights and indirect feedback of mainly used and lesser-used functionalities.

Contributions to the Literature

The contribution of our paper to the literature is manifold. We

- established Studierfenster (www.studierfenster.at): as a free, non-commercial open science client-server framework for (bio-)medical image analysis in the research community;
- provide a wide range of capabilities with our framework, like the visualization of medical data (CT, MRI, etc.) in two-dimensional (2D) and three-dimensional (3D) space in common web browsers, like Google Chrome, Mozilla Firefox, Safari or Microsoft Edge;
- conducted a usability user study with Studierfenster including medical and non-medical experts in (bio-)medical image analysis;
- report user study results in an ISO standard (ISO-Norm) questionnaire.

Author Contribution Conceived and designed the experiments: JE, DW, MW, CARB, FK, AP, MS, CD, DK, YJ, CG, JL, AP. Performed the experiments: JE, DW, MW, CARB, FK, AP, MS, CD, DK, YJ, CG, JL, AP. Analyzed the data: JE, DW, MW, CARB, FK, AP, MS, CD, DK, YJ, CG, JL, AP. Contributed reagents/materials/analysis tools: JE, DW, MW, CARB, FK, AP, MS, CD, DK, YJ, CG, JL, AP. Wrote the paper: JE.

Funding This work received funding from the Austrian Science Fund (FWF) KLI 678-B31: “enFaced: Virtual and Augmented Reality Training and Navigation Module for 3D-Printed Facial Defect Reconstructions” and the TU Graz Lead Project (“Mechanics, Modeling, and Simulation of Aortic Dissection”). Moreover, this work was supported by “CAMed” (COMET K-Project 871132), which is funded by the Austrian Federal Ministry of Transport, Innovation, and Technology (BMVIT); the Austrian Federal Ministry for Digital and Economic Affairs (BMDW); and the Styrian Business Promotion Agency (SFG). Finally, the REACT-EU project KITE (Plattform für KI-Translation Essen) also provided support.

Availability of Data and Materials/Code Availability All data and materials are available under Studierfenster (www.studierfenster.at) and Figshare (https://figshare.com/authors/_/3752557). All the software and source code of Studierfenster are licensed under the GNU General Public License (GPL) version 3. The rights of third-party components are held by their producers, with their respective license. Text and multimedia material, such as images and videos, are released under the Creative Commons Attribution-Sharealike 3.0 Unported license (CC-BY-SA) and the GNU Free Documentation License (GFDL) (unversioned, with no invariant sections, front-cover texts, or back-cover texts). Text, multimedia, and source code can be re-used and modified, given that the original sources are cited, namely, (1.) Studierfenster/Studierfenster.at and (2.) this whitepaper.

Declarations

Ethics Approval Not applicable.

Consent to Participate Not applicable.

Consent for Publication Not applicable.

Conflict of Interest The authors declare no competing interests.

References

1. Egger J. Pre- and Postoperative Segmentation and Virtual Stenting of Aneurysms and Stenosis. Dissertation in Computer Science (Dr. rer. nat.), Philipps-University of Marburg, Department of Mathematics and Computer Science. 215, 2009.
2. Akamatsu Y, et al. Intraoperative neuronavigation system without rigid pin fixation. *No Shinkei Geka*. 37(12):1193–9, 2009.
3. Ghose S, et al. A review of segmentation and deformable registration methods applied to adaptive cervical cancer radiation therapy treatment planning. *Artif Intell Med*. 64(2):75–87, 2015.
4. Abbas Q, et al. Skin tumor area extraction using an improved dynamic programming approach. *Skin Res Technol*. 18(2):133–42, 2012.
5. Zhang Y-J. *Advances in Image and Video Segmentation*. Hershey, PA: IRM Press. 457, 2006.
6. Goldenberg SL, Nir G, Salcudean SE. A new era: artificial intelligence and machine learning in prostate cancer. *Nature Reviews Urology*. 2019 Jul;16(7):391–403.
7. McDermott MB, Wang S, Marinsek N, Ranganath R, Foschini L, Ghassemi M. Reproducibility in machine learning for health research: Still a ways to go. *Science Translational Medicine*. 2021 Mar 24;13(586).
8. Egger J, Tokuda J, Chauvin L, Freisleben B, Nimsky C, Kapur T, Wells W. Integration of the OpenIGTlink network protocol for image guided therapy with the medical platform MeVisLab. *The international Journal of medical Robotics and Computer assisted Surgery*, 8(3):282–390, 2012.
9. Egger J, Kapur T, Fedorov A, Pieper S, Miller JV, Veeraraghavan H, Freisleben B, Golby AJ, Nimsky C, Kikinis R. GBM Volumetry using the 3D Slicer Medical Image Computing Platform. *Sci. Rep.* 3, 1364, 2013.
10. Wolf I, et al. The medical imaging interaction toolkit (MITK): a toolkit facilitating the creation of interactive software by extending VTK and ITK. *Proceedings Volume 5367, Medical Imaging 2004: Visualization, Image-Guided Procedures, and Display*, 2004.
11. Tian J, et al. A Novel Software Platform for Medical Image Processing and Analyzing. *IEEE Transactions on Information Technology in Biomedicine*, Volume: 12, Issue: 6, pp. 800–812, 2008.
12. Rosset A, et al. OsiriX: An Open-Source Software for Navigating in Multidimensional DICOM Images. *Journal of Digital Imaging*, Volume 17, Issue 3, pp 205–216, 2004.
13. Yushkevich PA, Piven J, Hazlett HC, Smith RG, Ho S, Gee JC, Gerig G. User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability. *Neuroimage*. 2006 Jul 1;31(3):1116–28.
14. Egger J, Grosskopf S, O'Donnell T, Freisleben B. A software system for stent planning, stent simulation and follow-up examinations in the vascular domain. In 2009 22nd IEEE International Symposium on Computer-Based Medical Systems 2009 Aug 2 (pp. 1–7). IEEE.
15. Paladini G, Azar FS. An extensible imaging platform for optical imaging applications. In *Multimodal Biomedical Imaging IV 2009 Feb 20 (Vol. 7171, p. 717108)*. International Society for Optics and Photonics.

16. Kuhnt D, Bauer MH, Egger J, Richter M, Kapur T, Sommer J, Merhof D, Nimsky C. Fiber tractography based on diffusion tensor imaging compared with high-angular-resolution diffusion imaging with compressed sensing: initial experience. *Neurosurgery*. 2013 Jan;72(0 1):165.
17. Drouin S, Kochanowska A, Kersten-Oertel M, Gerard JJ, Zelmann R, De Nigris D, Bériault S, Arbel T, Sirhan D, Sadikot AF, Hall JA. IBIS: an OR ready open-source platform for image-guided neurosurgery. *International journal of computer assisted radiology and surgery*. 2017 Mar 1;12(3):363–78.
18. Zhang L, Li J, Li P, Lu X, Shen P, Zhu G, Shah SA, Bennarmoun M, Qian K, Schuller BW. MeDaS: An open-source platform as service to help break the walls between medicine and informatics. *arXiv preprint arXiv:2007.06013*. 2020 Jul 12.
19. Gsaxner C, Pepe A, Wallner J, Schmalstieg D, Egger J. Markerless Image-to-Face Registration for Untethered Augmented Reality in Head and Neck Surgery. *MICCAI*, pp. 1–9, 2019.
20. Chen X, et al. Development of a surgical navigation system based on augmented reality using an optical see-through head-mounted display. *J Biomed Inform*, 55:124–31, 2015.
21. Egger J, et al. HTC Vive MeVisLab integration via OpenVR for medical applications. *PloS One* 12, e0173972, 2017.
22. Abid A, Abdalla A, Abid A, Khan D, Alfozan A, Zou J. An online platform for interactive feedback in biomedical machine learning. *Nature Machine Intelligence*. 2020 Feb;2(2):86–8.
23. Lösel PD, van de Kamp T, Jayme A, Ershov A, Faragó T, Pichler O, Jerome NT, Aadepe N, Bremer S, Chilingaryan SA, Heethoff M. Introducing Biomedisa as an open-source online platform for biomedical image segmentation. *Nature communications*. 2020 Nov 4;11(1):1–4.
24. Egger J, Gsaxner C, Pepe A, Li J. Medical Deep Learning-A systematic Meta-Review. *arXiv preprint arXiv:2010.14881*. 2020 Oct 28.
25. Gsaxner C, Roth P, Wallner J, Egger J. Exploit fully automatic low-level segmented PET Data for training high-level Deep Learning Algorithms for the corresponding CT Data. *Plos One* 14(3): e0212550, 2019.
26. 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.* 7:e773, pp. 1–83, 2021.
27. Aja-Fernández S, de Luis Garcia R, Tao D, Li X, editors. *Tensors in image processing and computer vision*. Springer Science & Business Media; 2009 May 21.
28. Eichorn J. *Understanding AJAX: Using JavaScript to create rich internet applications*. Prentice Hall; 2006 Aug 1.
29. Bedoya CA, Wild D, Pepe A, Gsaxner C, Li J, Egger J. A web client-based online DICOM browser and NRRD converter for Studierfenster. In *Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications* 2021 Feb 15 (Vol. 11601, p. 116010P). International Society for Optics and Photonics.
30. Egger J. Non-contrast and contrast-enhanced t1-weighted MRI scans from a healthy subject. *ResearchGate*, <https://doi.org/10.13140/RG.2.2.29916.08326>, 2018.
31. Lindner L, Kolodziej M, Egger J. Skull-stripped Contrast-Enhanced MRI Datasets. *Figshare*, <https://doi.org/10.6084/m9.figshare.7472>, 2018.
32. Lindner L, Narnhofer D, Weber M, Gsaxner C, Kolodziej M, Egger J. Using synthetic training data for deep learning-based GBM segmentation. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* 2019 Jul 23 (pp. 6724–9). IEEE.
33. Wild D, Weber M, Egger J. Client/server based online environment for manual segmentation of medical images. In *The 23rd Central European Seminar on Computer Graphics (CESCG)*, pp. 1–8, Apr. 2019.
34. Bhandari A, Koppen J, Agzarian M. Convolutional neural networks for brain tumour segmentation. *Insights into Imaging*. 2020 Dec;11:1–9.
35. Malik H, Fatema N, Iqbal A. *Intelligent Data-Analytics for Condition Monitoring: Smart Grid Applications*. Elsevier; 2021 Mar 26.
36. Castillo D, Lakshminarayanan V, Rodríguez-Álvarez MJ. *MR Images, Brain Lesions, and Deep Learning*. *Applied Sciences*. 2021 Jan;11(4):1675.
37. Egger J. GBM Datasets. *ResearchGate*, <https://doi.org/10.13140/RG.2.2.33331.73760>, 2017.
38. Schmied M, Pepe A, Egger J. A patch-based-approach for aortic landmarking. In *Medical Imaging 2021: Biomedical Applications in Molecular, Structural, and Functional Imaging* 2021 Feb 15 (Vol. 11600, p. 1160010). International Society for Optics and Photonics.
39. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*. 2012;25:1097–105.
40. Li Y, Alansary A, Cerrolaza JJ, Khanal B, Sinclair M, Matthew J, Gupta C, Knight C, Kainz B, Rueckert D. Fast multiple landmark localisation using a patch-based iterative network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* 2018 Sep 16 (pp. 563–571). Springer, Cham.
41. Pepe A, Li J, Rolf-Pissarczyk M, Gsaxner C, Chen X, Holzapfel GA, Egger J. Detection, segmentation, simulation and visualization of aortic dissections: A review. *Medical image analysis*. 2020 Oct 1;65:101773.
42. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial networks. *Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014)*. pp. 2672–2680.
43. Prutsch A, Pepe A, Egger J. Design and development of a web-based tool for inpainting of dissected aortae in angiography images. In *The 24th Central European Seminar on Computer Graphics (CESCG)*, pp. 1–8, May 2020.
44. Pepe A, Mistelbauer G, Gsaxner C, Li J, Fleischmann D, Schmalstieg D, Egger J. Semi-supervised Virtual Regression of Aortic Dissections Using 3D Generative Inpainting. In *International Workshop on Thoracic Image Analysis* 2020 Oct 8 (pp. 130–140). Springer, Cham.
45. Dionysio C, Wild D, Pepe A, Gsaxner C, Li J, Alvarez L, Egger J. A cloud-based centerline algorithm for Studierfenster. In *Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications* 2021 Feb 15 (Vol. 11601, p. 1160115). International Society for Optics and Photonics.
46. 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* 2021 Feb 15 (Vol. 11598, p. 115981Q). International Society for Optics and Photonics.
47. Kodym O, Li J, Pepe A, Gsaxner C, Chilamkurthy S, Egger J, Španěl M. SkullBreak/SkullFix-Dataset for automatic cranial implant design and a benchmark for volumetric shape learning tasks. *Data in Brief*. 2021 Apr 1;35:106902.
48. Campe G, von Pistracher K. Patient Specific Implants (PSI). In *Cranial Implant Design Challenge*, 2020 8 Springer Cham 1 9
49. Li J, Pepe A, Gsaxner C, Campe G, von JA. Egger baseline approach for autoimplant: the MICCAI 2020 cranial implant design challenge. In *Multimodal Learning for Clinical Decision Support and Clinical Image-Based Procedures*, 2020 Oct 4 Springer Cham 75 84
50. Morais R, Egger J, Alves V. Automated Computer-aided Design of Cranial Implants using a Deep Volumetric Convolutional Denoising Autoencoder. *WorldCist'19*, pp. 151–60, Mar. 2019.
51. Li J, Pimentel P, Szengel A, Ehlike M, Lamecker H, Zachow S, Estacio L, Doenitz C, Ramm H, Shi H, Chen X, et al. AutoImplant 2020-First MICCAI Challenge on Automatic Cranial Implant Design. *IEEE Transactions on Medical Imaging*. 2021 May 3.

52. Li J, Egger J. Towards the Automatization of Cranial Implant Design in Cranioplasty. Springer, Cham; 2020 Oct 8.
53. Jackson AS, Bulat A, Argyriou V, Tzimiropoulos G. Large pose 3D face reconstruction from a single image via direct volumetric CNN regression. In Proceedings of the IEEE international conference on computer vision 2017 (pp. 1031–9).
54. Karner F, Gsaxner C, Pepe A, Li J, Fleck P, Arth C, Wallner J, Egger J. Single-Shot Deep Volumetric Regression for Mobile Medical Augmented Reality. In Multimodal Learning for Clinical Decision Support and Clinical Image-Based Procedures 2020 Oct 4 (pp. 64–74). Springer, Cham.
55. Gsaxner C, Pepe A, Li J, Ibrahimasic U, Wallner J, Schmalstieg D, Egger J. Augmented Reality for Head and Neck Carcinoma Imaging: Description and Feasibility of an Instant Calibration, Markerless Approach. *Computer Methods and Programs in Biomedicine*. 2021 Mar 1;200:105854.
56. Wheeler G, Deng S, Toussaint N, Pushparajah K, Schnabel JA, Simpson JM, Gomez A. Virtual interaction and visualisation of 3D medical imaging data with VTK and Unity. *Healthcare technology letters*. 2018 Nov 12;5(5):148–53.
57. Egger J, Gall M, Wallner J, Boechat PD, Hann A, Li X, Chen X, Schmalstieg D. Integration of the HTC Vive into the medical platform MeVisLab. In *Medical Imaging 2017: Imaging Informatics for Healthcare, Research, and Applications 2017 Mar 13* (Vol. 10138, p. 1013817). International Society for Optics and Photonics.
58. Dice LR. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
59. Rockafellar RT, Wets RJ-B. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.
60. Weber M, Wild D, Wallner J, Egger J. A client/server based online environment for the calculation of medical segmentation scores. In 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) 2019 Jul 23 (pp. 3463–7). IEEE.
61. Taha AA, Hanbury A. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Med Imaging*. 2015 Aug 12;15:29. <https://doi.org/10.1186/s12880-015-0068-x>.
62. Molinari F, Zeng G, Suri JS. A state of the art review on intima-media thickness (imt) measurement and wall segmentation techniques for carotid ultrasound. *Computer methods and programs in biomedicine*, 100(3):201–221, 2010.
63. Vallieres M, Kay-Rivest E, Perrin LJ, Liem X, Furstoss C, Aerts HJ, Khaouam N, Nguyen-Tan PF, Wang CS, Sultanem K, Seuntjens J. Radiomics strategies for risk assessment of tumour failure in head-and-neck cancer. *Scientific reports*. 2017 Aug 31;7(1):1–4.
64. Chilamkurthy S, Ghosh R, Tanamala S, Biviji M, Campeau NG, Venugopal VK, Mahajan V, Rao P, Warier P. Development and validation of deep learning algorithms for detection of critical findings in head CT scans. *arXiv preprint arXiv:1803.05854*. 2018 Mar 13.
65. Wallner J, Mischak I, Egger J. Computed tomography data collection of the complete human mandible and valid clinical ground truth models. *Scientific Data* volume 6, Article number: 190003, 2019.
66. Masoudi M, et al. A new dataset of computed-tomography angiography images for computer-aided detection of pulmonary embolism. *Scientific Data*, 5:180180, 09 2018.
67. Menze BH, et al. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS), *IEEE Transactions on Medical Imaging*, vol. 34, nr. 10, pp. 1993–2024, Oct. 2015.
68. Armato SG, et al. The reference image database to evaluate response to therapy in lung cancer (rider) project: A resource for the development of change-analysis software. *Clinical Pharmacology & Therapeutics*, 84(4):448–456, 2008.
69. Nakamae Y, et al. Displacement of the large colon in a horse with enterolithiasis due to changed positions observed by computed tomography. *J Equine Sci*. 29(1): 9–13, 2018.
70. Vega F, et al. Image processing applied to measurement of particle size. *J. Phys.: Conf. Ser.* 582 012061, 2015.
71. Sedmak G. Image processing for astronomy. Meeting on Advanced Image Processing and Planetological Application, Vulcano, Italy, Sept. 16–18, 1985 Società Astronomica Italiana, *Memorie* (ISSN 0037-8720), vol. 57, no. 2, p. 149–71, 1986.
72. Forte M. Image Processing Applications in Archaeology: Classification Systems of Archaeological Sites in the Landscape, in: Andresen, J., T. Madsen and I. Scollar (eds.), *Computing the Past. Computer Applications and Quantitative Methods in Archaeology. CAA92*. Aarhus University Press, Aarhus, pp. 53–62, 1993.
73. Egger J, Gall M, Tax A, Ücal M, Zefferer U, Li X, von Campe G, Schäfer U, Schmalstieg D, Chen X. Interactive reconstructions of cranial 3D implants under MeVisLab as an alternative to commercial planning software. *PLoS ONE*, 12(3): e0172694, 2017.
74. Egger J, Wallner J, Gall M, Chen X, Schwenzer-Zimmerer K, Reinbacher K, Schmalstieg D. Computer-aided position planning of miniplates to treat facial bone defects. *PLoS One*. 2017 Aug 17;12(8):e0182839.
75. Memon AR, Wang E, Hu J, Egger J, Chen X. A review on computer-aided design and manufacturing of patient-specific maxillofacial implants. *Expert review of medical devices*. 2020 Apr 2;17(4):345–56.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Jan Egger^{1,2,3} · Daniel Wild^{1,2} · Maximilian Weber^{1,2} · Christopher A. Ramirez Bedoya^{1,2} · Florian Karner^{1,2} · Alexander Prutsch^{1,2} · Michael Schmied^{1,2} · Christina Dionysio^{1,2} · Dominik Krobath^{1,2} · Yuan Jin^{1,2,4} · Christina Gsaxner^{1,2} · Jianning Li^{1,2,3} · Antonio Pepe^{1,2}

¹ Institute of Computer Graphics and Vision, Faculty of Computer Science and Biomedical Engineering, Graz University of Technology, Inffeldgasse 16, 8010 Graz, Australia

² Computer Algorithms for Medicine Laboratory, Graz, Austria

³ Institute for Artificial Intelligence in Medicine, AI-guided Therapies, University Hospital Essen, Girardetstraße 2, 45131 Essen, Germany

⁴ Research Center for Connected Healthcare Big Data, ZhejiangLab, 311121, Hangzhou, Zhejiang, China