Automatic Skull Defect Restoration and Cranial Implant Generation for Cranioplasty

Jianning Li[a,h,d,g,*], Gord von Campe[b], Antonio Pepe[a,d,g], Christina Gsaxner[a,d,g], Enpeng Wang[e], Xiaojun Chen[e], Ulrike Zefferer[h], Martin Tödtling[h], Marcell Krall[h], Hannes Deutschmann[f], Ute Schäfer[h,g], Dieter Schmalstieg[a,g], Jan Egger[a,c,d,g,*]

[a]Institute of Computer Graphics and Vision, Graz University of Technology, Inffeldgasse 16, 8010 Graz, Austria
[b]Department of Neurosurgery, Medical University of Graz, Auenbruggerplatz 29, Graz, Austria
[c]Department of Oral and Maxillofacial Surgery, Medical University of Graz, Auenbruggerplatz 2, 8036 Graz, Austria
[d]Computer Algorithms for Medicine Laboratory, Graz, Austria
[e]School of Mechanical Engineering, Shanghai Jiao Tong University, Minhang District, 200240 Shanghai, China
[f]Department of Radiology, Medical University of Graz, Auenbruggerplatz 2, 8036 Graz, Austria
[g]BioTechMed-Graz, Mozartgasse 12/II, 8010 Graz, Austria
[h]Research Unit Experimental Neurotraumatology, Department of Neurosurgery, Medical University Graz, Auenbruggerplatz 2[2], 8036 Graz

ARTICLE INFO

ABSTRACT

A fast and fully automatic design of 3D printed patient-specific cranial implants is highly desired in cranioplasty – the process to restore a defect on the skull. We formulate skull defect restoration as a 3D volumetric shape completion task, where a partial skull volume is completed automatically. The difference between the completed skull and the partial skull is the restored defect; in other words, the implant that can be used in cranioplasty. To fulfill the task of volumetric shape completion, a fully data-driven approach is proposed. Supervised skull shape learning is performed on a database containing 167 high-resolution healthy skulls. In these skulls, synthetic defects are injected to create training and evaluation data pairs. We propose a patch-based training scheme tailored for dealing with high-resolution and spatially sparse data, which overcomes the disadvantages of conventional patch-based training methods in high-resolution volumetric shape completion tasks. In particular, the conventional patch-based training is applied to images of high resolution and proves to be effective in tasks such as segmentation. However, we demonstrate the limitations of conventional patch-based training for shape completion tasks, where the overall shape distribution of the target has to be learnt, since it cannot be captured efficiently by a sub-volume cropped from the target. Additionally, the standard dense implementation of a convolutional neural network tends to perform poorly on sparse data, such as the skull, which has a low voxel occupancy rate. Our proposed training scheme encourages a convolutional neural network to learn from the high-resolution and spatially sparse data. In our study, we show that our deep learning models, trained on healthy skulls with synthetic defects, can be transferred directly to craniotomy skulls with real defects of greater irregularity, and the results show promise for clinical use. Project page: https://github.com/Jianningli/MIA.

© 2021 Elsevier B. V. All rights reserved.

1. Introduction

Cranioplasty is the surgical process where a skull defect, caused in a brain tumor surgery or by trauma, is repaired

*Corresponding authors: Jianning Li (jianning.li@icg.tugraz.at) and Jan Egger (egger@tugraz.at)

using a cranial implant, which precisely replaces the missing cranial bone. Currently, after removal of the defective bone parts, computed tomography (CT) scans are obtained, and 3D implants are designed manually using computer-aided design (CAD) software, using the scans as reference. Often, this step requires access to expensive commercial CAD software. Even if free software alternatives are available, manual implant design is still time-consuming and requires trained specialists. The implant is manufactured by an external supplier based on the CAD model, which can take days or even weeks. In the meantime, the patient will require temporary protection of the brain, and later a second surgery – including repeated anesthesia – for the replacement of the implant.

The advent of bio-compatible 3D printing (Bose et al. (2019); Rengier et al. (2010); Park et al. (2019); Chia and Wu (2015); Friebe et al. (2018)) creates an opportunity to streamline this procedure and reduce the wait time: Theoretically, the implant could be manufactured during a single intervention, directly in the operation room (in-OR). However, to avoid undue extension of the surgical procedure, manual design of the implant is not practical, since it would take too long.

Therefore, this paper explores the option of automatic generation of an implant model from a CT scan using deep learning. Our contributions can be summarized as follows:

- We demonstrate that automatic cranial implant design can be accomplished in a fully data-driven manner without requiring shape constraints or priors. For this purpose, we formulated the problem as a volumetric shape completion task using a convolutional neural network (CNN).

- We construct a large skull database and show how data pairs can be generated automatically for supervised 3D shape learning without human annotation.

- We propose a patch-based training strategy which supports the learning of high-resolution and spatially sparse volumetric shapes.

- We investigate how skip connections can be used in an encoder-decoder network to improve 3D shape completion and propose a model ensembling technique to benefit from skip connections, while avoiding its adverse effects.

2. Related Work

Several CAD approaches exist for skull defect restoration and cranial implant reconstruction, which all exploit the apparent symmetry of the human skulls or rely on skull shape priors (Gall et al. (2016); Chen et al. (2017); Marzola et al. (2019); Egger et al. (2017); Kung et al. (2013); Fuessinger et al. (2017)). These approaches are however not optimal, considering that human skulls are not strictly symmetric. Moreover, these approaches are still time-consuming, limiting the applicability for instant manufacturing.

The work of Morais et al. (2019) is the closest to our study in that they also try to automate the design of cranial implant using deep learning. Compared to this work, our current study explores the following three major areas of improvement:

First, Morais et al. (2019) used low-resolution skull data (30^3, 60^3 and 120^3) for both training and evaluation of an autoencoder style network. However, real-life head CT scans used in clinical routines or for cranial implant design typically have a size of $512 \times 512 \times Z$, where Z is the number of axial slices. Training and testing a deep neural network on high resolution 3D images is challenging, as the memory requirement grows in a cubic manner. Our method trains a deep learning model on large 3D images of the skull with a standard desktop computer, which makes it easy and affordable to deploy the model at medical institutions.

Second, unlike most medical images, the skull images are binary and *sparse*, i.e., the valid (occupied) voxels, which represent the inner and outer skull surfaces, are sparsely distributed across the image volume. Therefore, conventional deep learning based methods, which proved successful in other medical images and tasks, may not be effective or efficient for our shape completion task. The high dimensionality and sparsity are not addressed in the work of Morais et al. (2019).

Third, the generalization ability of the deep learning models on various shapes and defect patterns is often the key considerations for shape completion tasks. For the cranial implant design task, the generalization ability is very relevant, as the skull defects can have various shapes, sizes and positions, depending on the specific underlying pathology (e.g., the size and location of a brain tumor) of a patient. In the work of Morais et al. (2019), simple synthetic defects with a cubic pattern were used for the evaluation of their method. We address the problem by injecting random defects for each healthy skull – a data augmentation technique that enables our deep learning models to generalize well to varied synthetic skull defects as well as to real craniotomy defects.

Skull defect restoration can be formulated as 3D volumetric shape completion, aiming at predicting the missing structures of the defected skull volume. In computer graphics, 3D shape completion has been intensively studied. Related approaches include classical mesh processing methods that directly operate on shapes represented as 3D triangular meshes (Kazhdan et al. (2006); Kazhdan and Hoppe (2013); Zhao et al. (2007); Ngo and Lee (2011); Sakr et al. (2018)). Some approaches complete the shape by exploiting the symmetry of 3D point clouds (Schiebener et al. (2016); Sung et al. (2015); Mitra et al. (2006)). The exploitation of symmetry for shape completion is similar to some of the interactive approaches for skull defect restoration (Angelo et al. (2019); Marzola et al. (2019); Gall et al. (2016); Egger et al. (2017); Chen et al. (2017)).

Data-driven approaches, especially deep learning approaches, also play an important role for 3D shape completion, facilitated by publicly available 3D shape datasets such as ShapeNet (Chang et al. (2015)). These approaches usually train a 3D CNN on a volumetric representation of a 3D point cloud of various objects such as cars, chairs, airplanes, etc. Most approaches use an encoder-decoder architecture for this purpose (Dai et al. (2016); Stutz and Geiger (2018); Han et al. (2017); Li et al. (2017); Litany et al. (2017)), operating
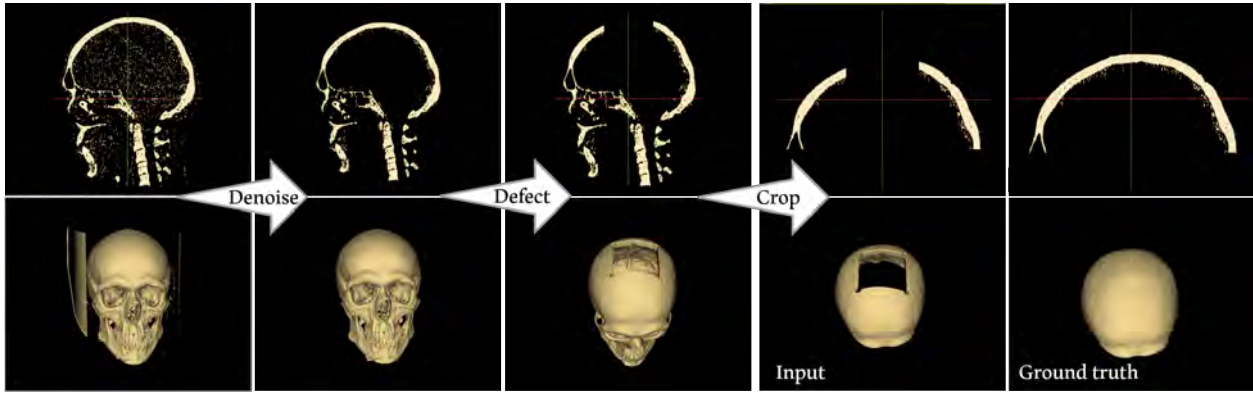
Fig. 1. Data preprocessing pipeline. From left to right: skull denoising using 3D connected component analysis; defect (hole) injection into the denoised skull in the neurocranium; skull cropping to keep only the neurocranial area, which is the region of interest (ROI) for cranial implant design.

on a binary voxel grid obtained from the 3D point cloud.

The binary voxel grid supports regular 3D convolution, but its high memory cost (compared to the sparse 3D point data) prohibits learning at high resolutions. For instance, the encoder-decoder network (Dai et al. (2016)) takes as input a voxel grid of only 32^3 voxels and produces a prediction of the same dimension, which is synthesized to a final output of 128^3. Han et al. (2017) uses voxel grids of 256^3. However, instead of directly processing the high-resolution voxel grid, they downsample the voxel grid to 32^3 and feed the low-resolution voxel grid into an encoder-decoder network to learn a *global* shape structure. In an additional step, they train another network using 3D patches (32^3) extracted around the defected area of the high-resolution voxel grid for shape refinement. These methods start from small point clouds, so a low-resolution voxel grid suffices to preserve the shape of the point cloud.

Other recent methods, which apply convolutional operations directly to point clouds for 3D shape classification and segmentation (Qi et al. (2016, 2017); Li et al. (2018a); Yang et al. (2017b); Li et al. (2018c); Liu et al. (2019)) are limited to small point clouds as well (Wu et al. (2014)).

Generative adversarial networks (GAN) also play a role in 3D shape completion (Wang et al. (2017); Wu et al. (2020); Sarmad et al. (2019)). Similar to the previously mentioned studies, GAN solutions are evaluated on small shapes from ShapeNet (Chang et al. (2015)). Specifically, Wang et al. (2017) build their methods primarily on a 3D Encoder-Decoder Generative Adversarial Network (3D-ED-GAN), which takes as input a coarse and corrupted voxel grid (32^3) and outputs a completed grid of the same dimension. A recurrent convolutional network is further adopted to increase the resolution of the completed grid to 128^3. Wu et al. (2020) and Sarmad et al. (2019) propose to perform shape completion directly on 3D point clouds, considering that point clouds represent the raw data structure of various scanning devices.

However, implant generation is a volumetric shape completion task on high-resolution volumes segmented from high-quality head CT scans ($512^2 \times 128$). At this scale, we found that going through an intermediate low-resolution step, as the voxel grid methods discussed above do, is neither efficient (too

slow) nor effective (low quality results). A similar restriction applies to the point-based methods mentioned above, which are not able to deal with millions of points, as would be required when converting high-resolution CT scans to a point representation. Another issue derived from the high resolution is the spatial sparsity of the skull data: The skull can be seen as a two-dimensional manifold embedded in a three-dimensional volumetric space, where most of the voxels in the image volume are unoccupied (Figure 1). Therefore, unlike previous methods, our method performs 3D shape completion directly on the sparse regions of the high-resolution volumetric data.

3. Materials

3.1. Datasets

We constructed a database containing 167 healthy skulls. The skulls are segmented manually by neurosurgeons from head CT scans, which are acquired in clinical routine at a high resolution of $512 \times 512 \times Z$ (Z ranges from 255 to 480). As bone density changes with age, an individualized Hounsfield units segmentation threshold was used for each skull, to ensure that the complete skull anatomy was preserved after the segmentation process. The resulting datasets thus have the complete bone structures of each skull without a defect.

3.2. Preprocessing and Data Pair Creation

The overall data preprocessing pipeline (Figure 1) consists of three main steps: denoising, hole injection and cropping. The denoising step removes the noise and artifacts (e.g., the head holder of the CT table) by using 3D connected component analysis. We create training pairs out of the skull data by injecting defects (holes) into the neurocranial area of the healthy skulls. The process is to simulate defects resulting from a craniotomy surgery, where holes are drilled in the corners of the bone flap to be removed. Afterwards, the data driven approach will learn to refill the defects based on these data pairs (i.e., the skull with a hole injected and the corresponding healthy skull). This course of action has two main advantages: (1) When injecting a hole into a healthy skull, the ground truth (i.e., the healthy skull) for reconstruction is known. (2) We
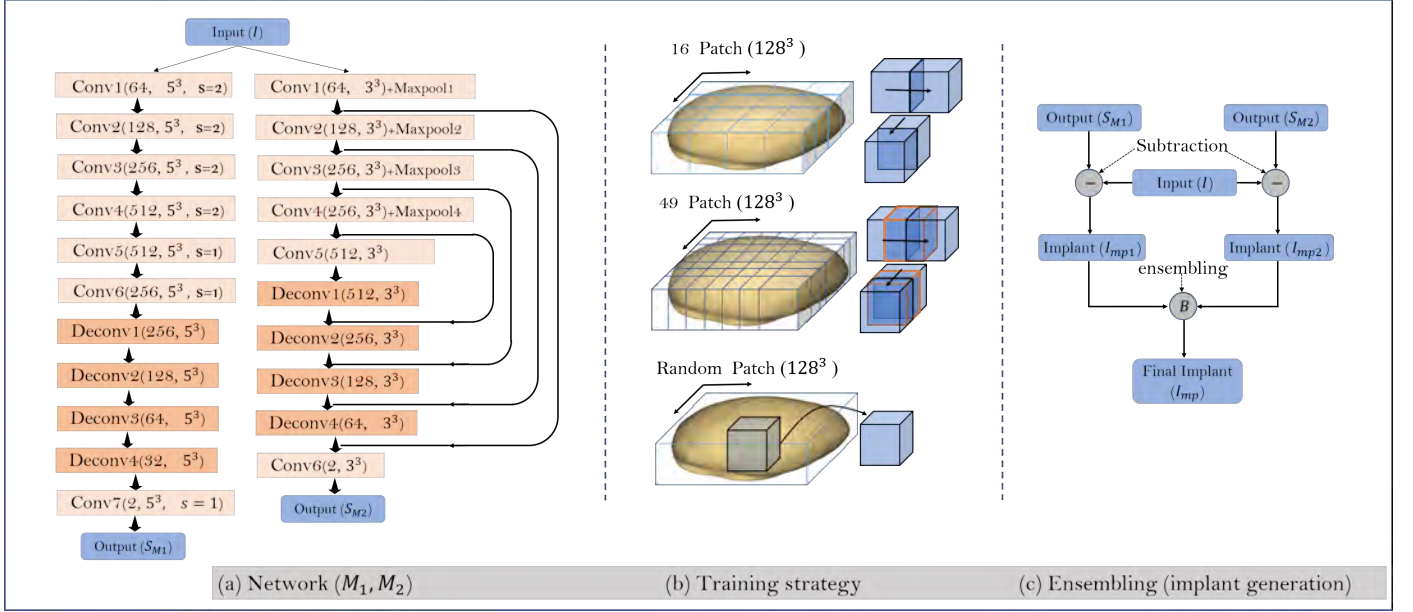
Fig. 2. Method Overview: (a) the network configurations of M_1 and M_2, **(b)** the training strategy − non-overlapping cropping (top), overlapping cropping (middle), random cropping (bottom), **(c)** model ensembling based on bounding box for cranial implant generation.

can create multiple data pairs out of a single healthy skull by injecting holes with different positions, sizes and shapes.

In order to create realistic defects in the healthy skulls, we refer to the real morphology of skull defects that are manually created by neurosurgeons in a craniotomy, according to a published craniotomy dataset (Gall et al. (2019)). As is shown in Figure 1, the artificially injected defects used in the datasets are simplified, but represent the overall morphology of real defects after a craniotomy (the real defects have rougher borders compared to the artificial ones). For each healthy skull, we injected nine random defects with different positions, sizes and shapes, resulting in a total of $167 \times 9 = 1503$ data pairs (nine defected skulls per one healthy skull). The defected skull in Figure 1 is just one of the nine random situations where the defect is injected around the middle. The data pairs are further split into a training set of 765 (85×9) pairs and a test set of 738 (82×9) pairs. Since the region of interest (ROI) for cranial implant design in neurosurgery is the neurocranial area, the last data preprocessing step is to crop the skull and discard the facial area. The cropping results in a uniform skull dimension of $512 \times 512 \times 128$ [1]. Besides the 167 healthy skulls with artificial defects, we also collected one head CT from a craniotomy case for the evaluation of the approach in a clinical setting.

4. Method

4.1. Problem Formulation and Network Architecture

Restoring defects in the skull can be formulated as a 3D volumetric shape completion task, where the input skull S_d is represented as a 3D volume with a missing part, and the output

S_c is the completed skull with the missing bone restored. Both S_d and S_c are binary, where 1 represents voxels belonging to the skull and 0 represents unoccupied voxels (the background and the empty space inside the skull) in the image volume. Intuitively, the cranial implant I_{mp} can be expressed as the subtraction of S_d from the completed skull S_c:

$$I_{mg} = S_c - S_d \tag{1}$$

Equation (1) can be rewritten as:

$$S_c = S_d + I_{mg} \tag{2}$$

According to Equation (2), skull shape completion can be divided into two subtasks, the reconstruction of the original structure of the skull (i.e., the input S_d) and the restoration of the missing skull bone I_{mp}, which should be fulfilled simultaneously by a CNN. In our study, two encoder-decoder style networks, shown in Figure 2 (a) are employed, which are referred to as M_1 (left, without skip connections) and M_2 (right, with skip connections), respectively. For M_1, the encoder is comprised of four convolutional layers with a stride of two for down-sampling and two additional convolutional layers with a stride of one for feature embedding (i.e., *Conv5* and *Conv6*). The decoder is comprised of four deconvolutional layers for up-sampling, and the output layer is a convolutional layer with a stride of one. The total number of trainable parameters of M_1 is 82.076 million. For M_2, max-pooling is used in the down-sampling path, and skip connections are added between corresponding down-sampling and up-sampling layers. The number of trainable parameters of M_2 is 41.024 million. To simplify notation, the input of M_1 and M_2 is denoted as I, and their output is denoted as S_{M1} and S_{M2}, as in Figure 2 (a).

---

[1] We selected the upper 128 axial slices of the skull, which approximately represents the neurocranium.

For the training of both networks, a Dice based loss function (Yang et al. (2017a)) is used:

$$\mathcal{L}(y_p, y_g) = -2 \sum_{i=0}^{1} \frac{\sum y_p^i \circ y_g^i}{\sum y_p^i \circ y_p^i + \sum y_g^i \circ y_g^i}, \quad (3)$$

where $\circ$ denotes the Hadamard product (element-wise multiplication) of two 3D matrices. $\sum$ denotes the summation of matrix elements. y_p is the softmax output of the network, and y_g is the one-hot representation of the ground truth skull. Both y_p^i and y_g^i have a dimension $128 \times 128 \times 128$, which is the input/output size of the networks, as will be discussed in Section 4.3. As the final output is binary, y_p^0 and y_g^0 represent the background and y_p^1 and y_g^1 represent the foreground.

4.2. Model Ensembling for Cranial Implant Generation

According to Equation (1), the implant from M_1 and M_2 can be expressed as $I_{mp1} = S_{M1} - I$ and $I_{mp2} = S_{M2} - I$. We further propose an ensembling technique based on bounding boxes to combine the implants from each individual model into the final cranial implant. The bounding box is calculated using the implant from M_2, which is applied to the implant from M_1. The process is illustrated in Figure 2 (c). The bounding box

$$\mathbb{B}(x, y, z) = \begin{cases} 1, & \text{if } x \in [x_1, x_2], y \in [y_1, y_2], z \in [z_1, z_2] \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

is calculated according to the criteria $(5) - (7)$.

$$\begin{cases} X_p(x_1) \neq 0, X_p(x_1 - 1) = 0 \\ X_p(x_2) \neq 0, X_p(x_2 + 1) = 0 \\ X_p = \sum_{y=1}^{Y} \sum_{z=1}^{Z} I_{mp2} \end{cases} \quad (5)$$

$$\begin{cases} Y_p(y_1) \neq 0, Y_p(y_1 - 1) = 0 \\ Y_p(y_2) \neq 0, Y_p(y_2 + 1) = 0 \\ Y_p = \sum_{x=1}^{X} \sum_{z=1}^{Z} I_{mp2} \end{cases} \quad (6)$$

$$\begin{cases} Z_p(z_1) \neq 0, Z_p(z_1 - 1) = 0 \\ Z_p(z_2) \neq 0, Z_p(z_2 + 1) = 0 \\ Z_p = \sum_{y=1}^{Y} \sum_{x=1}^{X} I_{mp2} \end{cases} \quad (7)$$

where X, Y, Z are the numbers of axial, sagittal and coronal slices in I_{mp1}, I_{mp2}, and $\mathbb{B} \in \mathbb{R}^{X \times Y \times Z}$ ($X = Y = 512$, $Z = 128$). The final implant is given by:

$$I_{mp} = \mathbb{B} \cdot I_{mp1} \quad (8)$$

4.3. Training Strategy

M_1 and M_2 are trained separately, since they are designed to be complementary and not adversarial. The strategy for training the two networks is designed specifically for a volumetric shape completion task on high dimensional, spatially sparse data.

High Dimensionality. Patch-based training is a widely used strategy in deep learning when the data dimensionality is high (Yang et al. (2017a); Wang et al. (2019); Heinrich et al. (2019); Dou et al. (2017); Li et al. (2018b); Kamnitsas et al. (2017)). In a conventional patch-based training scheme, the deep learning model trains on a sub-volume randomly cropped from the data for each training epoch, which makes it difficult for the model to capture the overall characteristics of the complete volume. This disadvantage is tolerable when the overall shape of the target is not the primary concern. For instance, in tasks such as segmentation, the model only needs to learn to differentiate between voxels in the target and voxels in the background.

However, the disadvantage becomes less tolerable for tasks where the primary goal is to learn the overall shape of the target, such as 3D shape completion, as a small patch does not carry enough information about the overall shape in the volume. In skull shape completion, increasing the patch size could lead to improvements, but it is not a practical solution when the computational resources are limited.

As an alternative, we propose that training on patches sequentially cropped from the same data for consecutive epochs can encourage the network to learn the overall skull shape, compared to the random cropping method. In Figure 2 (b) top, the skull shape is divided into $4 \times 4 = 16$ equally-sized patches of dimension 128^3, and the patches are sequentially fed into the network for 16 consecutive epochs [2]. When another skull is selected, the same procedure is repeated.

By doing so, the whole skull shape can be involved in the optimization, before another skull is selected, so that the overall shape distribution of the skull can be captured by the deep learning model. The training strategy imposes a *virtual* skull shape prior in the shape learning process by telling the network, in an implicit manner, what each particular skull shape is like.

Spatial Sparsity. Another issue is the sparsity. The skull data are spatially sparse and can be seen as a two-dimensional manifold embedded in three-dimensional volumetric space ($512 \times 512 \times 128$). The overall voxel occupancy rate of a skull in the volumetric space is usually no more than 10 percent. While processing sparse data, most of the time, the convolution is operating on empty space, which causes inefficient usage of computational resources and difficulties for feature learning. Based on the fact that the level of sparsity is negatively related to the patch size, we propose that the learning difficulty posed by sparsity could be overcome by cropping more patches from the same skull, e.g., cropping patches in an overlapping manner, so that the *relative* patch size can be reduced [3]. In Figure 2 (b) middle, we crop $7 \times 7 = 49$ patches of dimensional 128^3 from a skull, and the patches are overlapped in the middle.

For ease of reference, we denote the model trained using non-overlapping cropping ($-n$), overlapping cropping ($-o$) and random cropping ($-r$) strategy as $M - n$, $M - o$ and $M - r$. For example, $M_1 - n$ denotes the M_1 trained using the

---

[2] Limited by the current GPU memory, the batch size is set to 1.

[3] The *relative* patch size is the size of the whole volume divided by the number of patches.

Table 1. Mean values of the Dice Similarity Score (DSC), the Jaccard Similarity Coefficient (JSC), Precision and Recall of the $82 \times 9 = 738$ test cases for the skulls and implants.

methods	skull				implant			
	DSC	*JSC*	*Precision*	*Recall*	*DSC*	*JSC*	*Precision*	*Recall*
$M_1 - n$	0.8861	0.7971	0.9023	0.8726	0.7276	0.5830	0.7042	0.7739
$M_1 - o$	0.8931	0.8086	0.9020	0.8863	0.7267	0.5810	0.6888	**0.7838**
$M_1 - r$	0.8598	0.7559	0.8573	0.8655	0.6080	0.4457	0.5437	0.7210
$M_2 - n$	**0.9508**	**0.9075**	**0.9622**	**0.9403**	0.7633	0.6257	**0.8059**	0.7346
ensembling	-	-	-	-	**0.7725**	**0.6396**	0.7847	0.7736
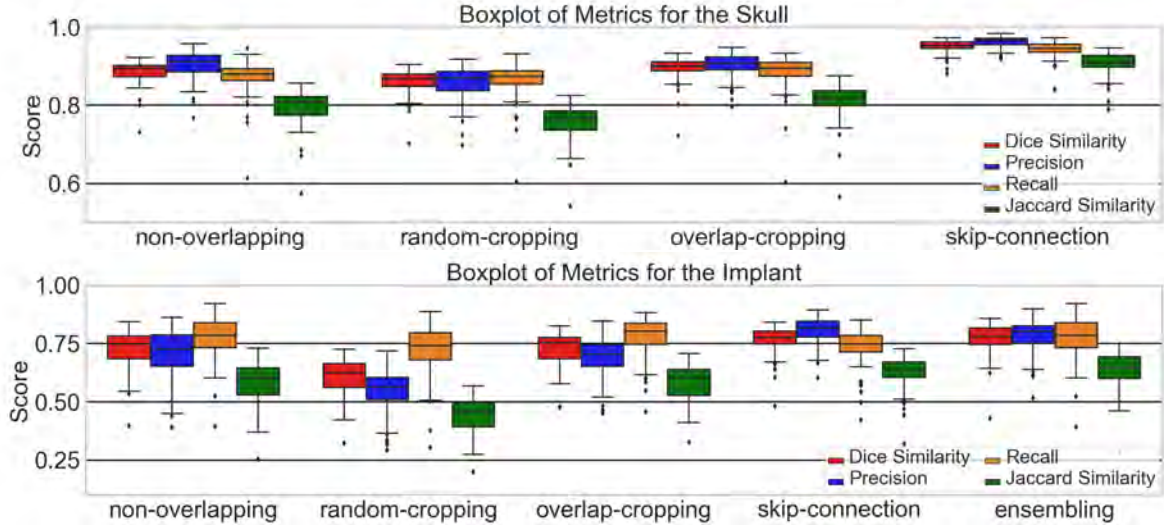
Fig. 3. Boxplot of the metrics for the skulls and implants by non-overlapping cropping ($M_1 - n$), random cropping ($M_1 - r$), overlapping cropping ($M_1 - o$), skip connection ($M_2 - n$) and ensembling ($M_1 - n$ and $M_2 - n$).

non-overlapping strategy. We will demonstrate that different training strategies can lead to significant differences in the maximum learning capacity a deep learning model can reach, when trained to their full convergence, for shape learning problems such as volumetric shape completion.

5. Experiments

5.1. Implementation

We implemented M_1 and M_2 using Tensorflow, on a machine with an Intel(R) Core(TM) i5-6600K CPU and a Nvidia GeForce GTX 1070Ti GPU. The dice loss function is defined in Equation (3), and the optimizer is Adam. The training ($85 \times 9 = 765$) and testing ($82 \times 9 = 738$) sets are described in Section 3. For the experiments, we train M_1 using the non-overlapping cropping, overlapping cropping and random cropping strategy and obtain the trained models $M_1 - n$, $M_1 - o$ and $M_1 - r$. M_2 is trained using non-overlapping cropping, and the trained model is $M_2 - n$. Each training takes approximately 150h for the model to fully converge (i.e., reach their maximum learning capacity). The purpose of the experimental design is to compare the training strategies (based on $M_1 - n$, $M_1 - o$, $M_1 - r$) and evaluate the two models (based on $M_1 - n$ and $M_2 - n$)

separately. For a fair comparison, all the models are trained from scratch (weights initialized from normal distribution) with exactly the same settings until full convergence, as monitored from the training loss/epoch curve. Full convergence means that the models have been trained to their full learning capacity on the training set. After training the models, the skull shape completion is also done in a patch-wise manner. After skull shape completion, each trained model generates the implants according to Equation (1). To evaluate the ensembling based method, $M_1 - n$ and $M_2 - n$ are selected to jointly produce an implant for each test case based on Equation (8).

5.2. Evaluation Metrics

We evaluated the trained models in terms of Dice Similarity Score (DSC) and Jaccard Similarity Coefficient (JSC, also known as Intersection Over Union (IOU)):

$$DSC = 2|G \cap P|/(|G| + |P|) \tag{9}$$

$$JSC = |P \cap G|/(|P| + |G| - |P \cap G|) \tag{10}$$

where P represents the prediction, and G represents the ground truth. These metrics are commonly used to measure the similarity between two binary volumetric shapes.

For voxel-level evaluation, the performance of the models to distinguish between unoccupied voxels (the background and the empty space inside the skull) and occupied voxels (voxels belonging to the skull) is measured using precision (also known as PPV, positive predictive value) and recall (also known as sensitivity, true positive rate):

$$PPV = TP/(TP + FP) \tag{11}$$

$$Recall = TP/(TP + FN) \tag{12}$$

where TP, FP, FN represent true positive, false positive and false negative, respectively. From this perspective, the models are considered as binary classifiers. For a more advanced evaluation, we extracted the surface model (triangular mesh) from some selected test cases and calculated the signed distance between the prediction and the ground truth. The signed distance is depicted using a 3D colormap. For a qualitative evaluation and comparison, we also give illustrations of some reconstructed skulls and implants in both 2D and 3D.

5.3. Results

We first performed skull reconstructions on the $82 \times 9 = 738$ test cases using the trained models $M_1 - n$, $M_1 - o$, $M_1 - r$ and $M_2 - n$. For each test case, we obtained the implant by subtracting the defected skull from the reconstructed skull according to Equation (1). We calculate the above mentioned metrics for both the skull (the reconstructed skull produced by M_1 and M_2 against the ground truth skull) and the implant (the implant obtained by each individual model as well as by ensembling against the ground truth implant). The results are shown in Table 1.

We observe from Table 1 that $M_1 - n$ and $M_1 - o$ outperform $M_1 - r$ regarding all the metrics for both the skull and implant, showing the superiority of the proposed training strategy ($-n$ and $-o$) compared to the conventional random cropping ($-r$) with respect to our shape completion task. $M_2 - n$ has the best performance compared to each individual model, and all the metrics regarding skull reconstruction exceed 0.90 (DSC: 0.9508, Precision: 0.9622, Recall: 0.9403, JSC: 0.9075), which is a significant improvement compared to M_1, even if M_2 is only half the size of M_1 in terms of the number of trainable parameters. However, despite the high accuracy for skull reconstructions, the scores for the implants remain mediocre, which could imply that the improvement is mainly due to the enhanced ability of M_2 to reconstruct the original (defective) structure of the skull rather than to restore the missing part, i.e., the implant, thanks to the use of skip connections. We will further discuss the effects skip connections have in our skull shape completion task in Section 5.5 and 5.6, where the reconstructed skulls and implants are visualized.

For all the models, the implant accuracy is lower than that of the skull, which can be attributed to three factors. First, the failure of the deep learning models when the defect is very large, in particular if the defect is created close to the lower border of the cropped skull and the contextual information is missing. Second, the implant is obtained through subtraction according to Equation (1), so that the errors from

skull reconstruction are accumulated in the implants. Third, and most importantly, most of the errors are located in uncritical regions, while the clinically relevant parts of the implant, i.e., implant boundaries, shape and curvatures, are of a relatively high standard.

The implants obtained from the ensembling of $M_1 - n$ and $M_2 - n$ according to Equation (8) are also quantitatively evaluated against the ground truth. The scores are shown in Table 1 (last row). Regarding DSC and JSC, we can see that the implants produced by the ensembling based method are superior to the implants produced by $M_1 - n$ or $M_2 - n$ alone. The improved implant quality due to ensembling can be better observed from qualitative illustrations (Section 5.6).

The scores of the test cases are also given as boxplots in Figure 3. Unlike the accuracy values in Table 1, which are calculated by taking the average of the $82 \times 9 = 738$ test scores for each metric, each boxplot shows the accuracy values of the 82 test cases. As we have considered nine different defects for each skull in the test set, we take the average of the nine test scores as the final score for each of the 82 skulls and implants for the boxplots.

5.4. Statistical Significance Analysis

Via t-tests, we further analysed whether the improvement of using skip connections ($M_2 - n$) and the proposed training strategy ($-n$, $-o$) is statistically significant regarding all the metrics for both the skull and the implant. We calculated:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{s_p \sqrt{2/n}} \tag{13}$$

where $s_p = \frac{1}{2} \sqrt{s_{X_1}^2 + s_{X_2}^2}$ and $\bar{X}_1$, $\bar{X}_2$, $s_{X_1}^2$, $s_{X_2}^2$ are the mean and the estimated variance with respect to a metric from two trained models. n is the number of test cases. Table 2 shows the results.

We adopt the commonly used $P_T = 0.05$ as the threshold to categorize the difference as significant or insignificant. Table 2 shows that $M_2 - n$ performs significantly better than $M_1 - o$, $M_1 - n$ and $M_1 - r$ for both the skull and the implant regarding most of the metrics. The proposed training strategy ($M_1 - n$ and $M_1 - o$) also achieves significant improvements compared to a conventional training strategy ($M_1 - r$).

5.5. Visualizations

The skull shape completion results of different skulls with defects of different shapes, positions and sizes are shown in 3D and 2D in Figure 4 and Figure 5, respectively. The first to last columns of Figure 4 show the input, ground truth and skull shape completion results of $M_1 - o$, $M_1 - n$, $M_1 - r$ and $M_2 - n$, respectively. Judging from the 3D illustration in Figure 4, it is evident that the reconstruction by $M_1 - r$ is of the worst quality both in terms of reconstructing the original structure and of restoring the missing part. $M_1 - o$ and $M_1 - n$ can reconstruct a complete skull with sufficient quality, but the shape of the reconstructed skull still shows some distortion from that of the ground truth. Due to the mismatch, the implant obtained by subtracting the input from the reconstruction will contain some

Table 2. P values ($P_T = 0.05$) among the evaluation metrics of the four approaches for statistical significance analysis.

	skull				implant			
	DSC	*Precision*	*Recall*	*JSC*	*DSC*	*Precision*	*Recall*	*JSC*
$M_2 - n \leftrightarrow M_1 - o$	$1.518e^{-34}$	$3.730e^{-35}$	$2.280e^{-18}$	$1.686e^{-38}$	$5.000e^{-4}$	$1.011e^{-18}$	$8.486e^{-5}$	$3.000e^{-4}$
$M_2 - n \leftrightarrow M_1 - n$	$4.511e^{-40}$	$7.456e^{-30}$	$5.481e^{-26}$	$4.330e^{-44}$	$1.400e^{-3}$	$2.240e^{-12}$	$2.300e^{-3}$	$1.100e^{-3}$
$M_2 - n \leftrightarrow M_1 - r$	$4.796e^{-55}$	$6.802e^{-48}$	$1.015e^{-28}$	$2.888e^{-60}$	$4.770e^{-30}$	$2.112e^{-50}$	$3.464e^{-1}$	$4.848e^{-34}$
$M_1 - n \leftrightarrow M_1 - r$	$6.645e^{-08}$	$1.164e^{-11}$	$2.902e^{-1}$	$1.294e^{-08}$	$2.308e^{-17}$	$9.265e^{-20}$	$6.000e^{-4}$	$2.380e^{-19}$
$M_1 - o \leftrightarrow M_1 - r$	$2.986e^{-11}$	$1.596e^{-12}$	$2.700e^{-3}$	$2.127e - 12$	$1.860e^{-18}$	$1.079e^{-19}$	$3.650e^{-05}$	$1.659e^{-20}$
$M_1 - o \leftrightarrow M_1 - n$	$1.221e^{-1}$	$9.448e-1$	$4.380e^{-2}$	$9.720e^{-2}$	$9.411e^{-1}$	$3.116e^{-1}$	$4.484e^{-1}$	$8.839e^{-1}$
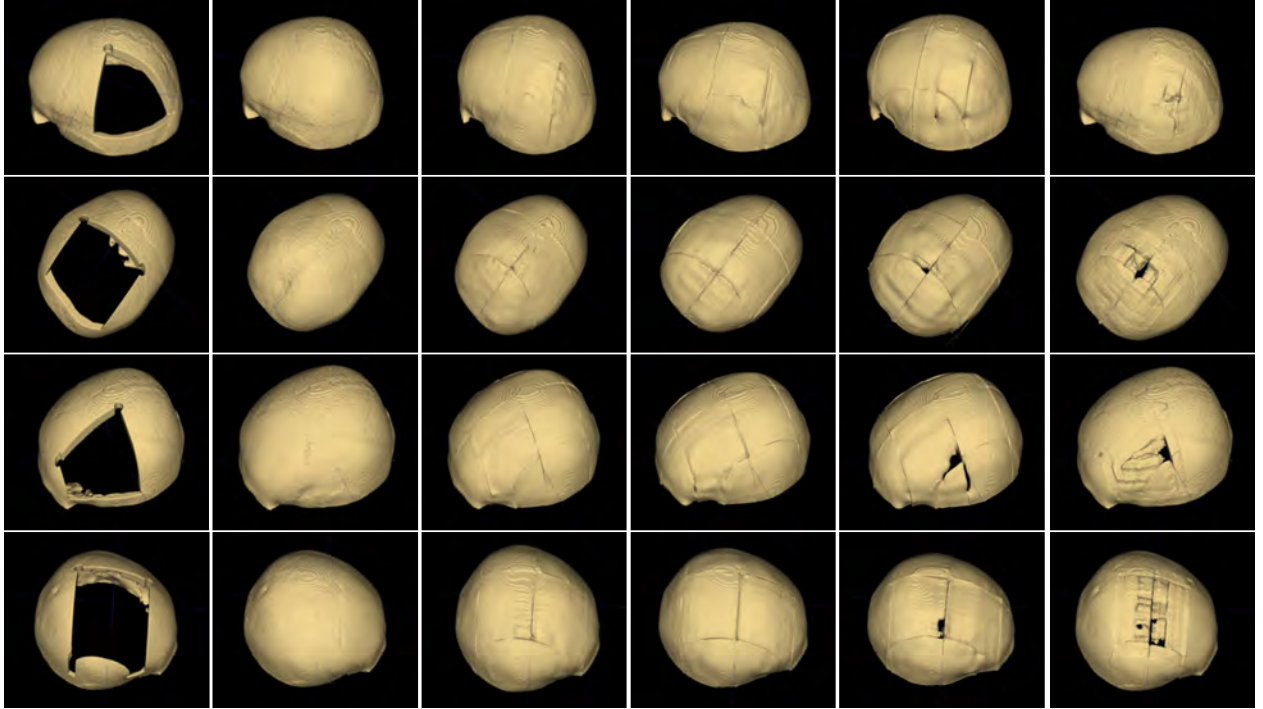
Fig. 4. 3D Visualization of skull shape completion results. Columns from left to right: the input, the ground truth, the completion results from $M_1 - o$, $M_1 - n$, $M_1 - r$, and $M_2 - n$.

noise around the borders (e.g., first row, Figure 7), which is one of the causes of the mediocre scores for the implant in Table 1.

The reconstruction results of M_2 are quite different from those of M_1 in that M_2 can reconstruct the original (defective) structure of the skull well. At the same time, we observe a *non-closed* restoration of the missing part, as can be seen from the last column of Figure 4. Note that, even if such failures can also be seen in the results of some random cases from $M_1 - r$ (fifth column, Figure 4), they are mainly caused by the reconstruction errors of the deep learning models. Nonetheless, the *non-closed* skull reconstruction is prevalent across all the test set for M_2, which could be attributed to the use of skip connections.

Note that the *stitching* effect on the surface of the skulls produced by M_1 is caused by the patch-wise training and inference scheme, where the output 3D patches are stitched together to form a complete skull during inference. It shows that the contacting borders of the output patches for a skull are not fully in congruency. However, the stitching effect is less noticeable on the surface of the skulls reconstructed by M_2.

Besides the 3D visualizations, we show in Figure 5 an overlay of the completed skull onto the original defected skull in the 2D sagittal and axial views to better illustrate how the restored missing part of the skull can match with the defected area. Matching shape and boundary is the primary concern when assessing the clinical usability of a cranial implant in cranioplasty. For cosmetic and aesthetic considerations, the implant is required to restore the original/normal shape of the patient's skull. In particular, the inner and outer curvatures and contact points of the implant should be largely in agreement with the original defected skull, so that the skull surface appears *continuous* around the implanted area. From Figure 5, we can see that the restored skull bone from $M_1 - o$, $M_1 - n$, $M_1 - r$, and $M_2 - n$ can connect with the original skull structure smoothly on both inner and outer surfaces. For $M_1 - r$, some discontinuities can be seen on the axial view of the restored skull bone (third column, Figure 5).

We can observe from Figure 5 that the restored part (i.e., the implant) can fit precisely against the borders of the skull
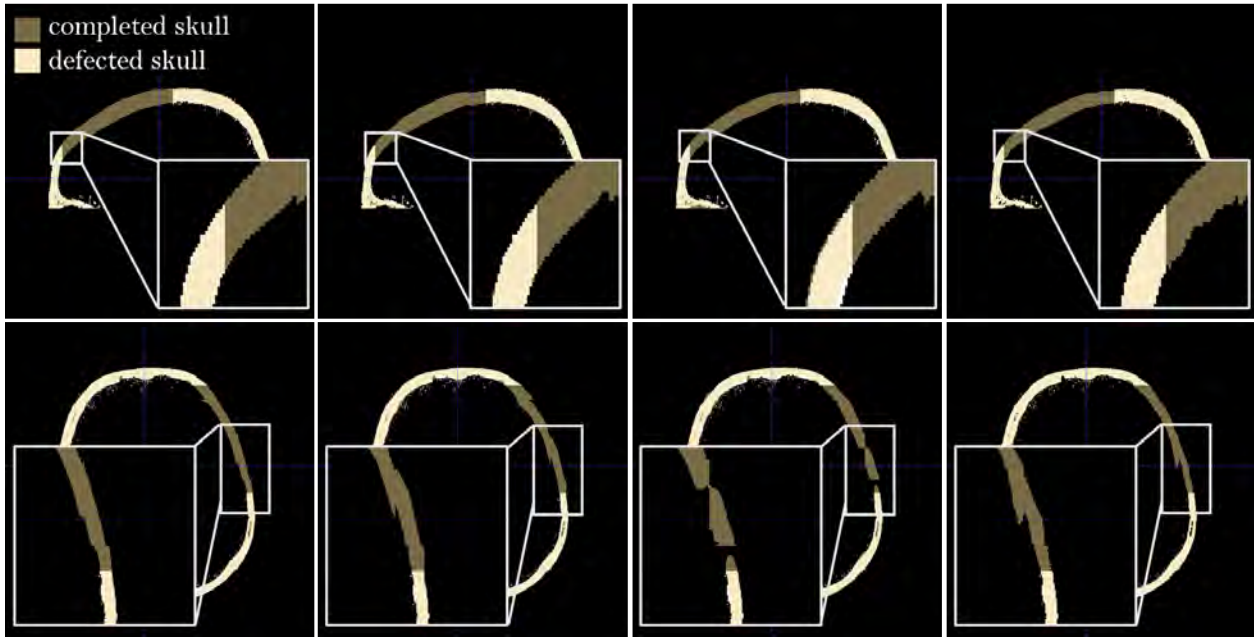
Fig. 5. A visual inspection of the completed skull overlaid onto the original defected skull in 2D sagittal (top row) and axial (bottom row) view, where the contacting parts (boundaries) between the defected area and the restored bone are zoomed in. From left to right: the completion results from $M_1 - o$, $M_1 - n$, $M_1 - r$, and $M_2 - n$

defect. However, a clinically usable cranial implant does not necessarily need to fit tightly in such a way. Instead, tiny gaps are allowed, since the borders of the defect are usually not sharp due to bone generation and scarring (von Campe and Pistracher (2020)). Even commercially designed and manufactured cranial implants are sometimes subject to manual intra-operative post-processing of the boundaries to make them fit. In our study, however, we considered the tight boundary contact, as shown in Figure 5, to be a desirable characteristic of our automatic cranial implant design algorithm, since a slightly too large (non-metal) implant can be further post-processed by the neurosurgeons when necessary, whereas no correction is possible if the automatically generated implant is too small.

5.6. Model Ensembling for Implant Generation

In this section, we will further illustrate how skip connections in an encoder-decoder network affect the volumetric shape completion task, and how the proposed model ensembling technique can take the advantages of M_2, while avoiding its adverse effects for implant generation. Besides $M_2 - n$, a trained model from M_1 is needed to generate implants via model ensembling according to Equation (8). Here, we chose $M_1 - n$. From the four reconstructed skulls of $M_2 - n$ and $M_1 - n$ as well as from the ground truth skulls in Figure 4, we extract the mesh models and calculate the signed distance field between the reconstruction and the ground truth meshes. The signed distance fields are represented by the 3D colormaps, as shown in Figure 6, which illustrate how the reconstruction errors are distributed across a skull. For all the colormaps in Figure 6, the color ranges are adjusted to [-7, 11], where the color center is set to 0 (green). By examining the 3D distance colormaps in the first row, we see that there is a mismatch (non-zero distance) between the reconstruction of $M_1 - n$ and the

ground truth throughout the skull. The mismatch is the cause of noise in the implant obtained through subtraction, as can be seen from the first row of Figure 7.

The distance colormap of $M_2 - n$ (in the second row) has properties distinct from that of $M_1 - n$. First, the distance is close to zero (shown as green), except in the restored region, which means that $M_2 - n$ achieves high accuracy in reconstructing the original skull structure, and the reconstructed skull can match precisely with the ground truth and the input, except in the missing part. The precise match can lead to an implant with clean borders, as can be seen from the second row of Figure 7. Second, the implant from $M_2 - n$ is *non-closed* on the surface, despite having clean borders.

The above mentioned distinct behaviors of M_1 and M_2 have been observed in all the 738 test cases, which we attribute to the use of skip connections. In our skull shape completion task, skip connections can enhance the ability of an encoder-decoder network to reconstruct the original defective skull structure, but the ability to restore the missing part may be undermined.

Figure 7, last row, shows the implants obtained from the ensembling of the two models based on the method described in Section 4.2. We can see that the ensembled implants have both *closed* surfaces and clean borders and the visual quality is superior to implants obtained from M_1 or M_2 alone.

Figure 8 shows the 3D signed distance colormaps for the implants obtained from $M_1 - n$ (left), $M_2 - n$ (middle) and the ensembling of $M_1 - n$ and $M_2 - n$ (right). From these colormaps, we can see that the boundaries as well as the small roundish corners of the implant have the smallest errors (the distance to the ground truth is close to 0) compared to other parts of the implant. The implant can fit precisely against the borders of the missing region on the input skull, which is a desirable property
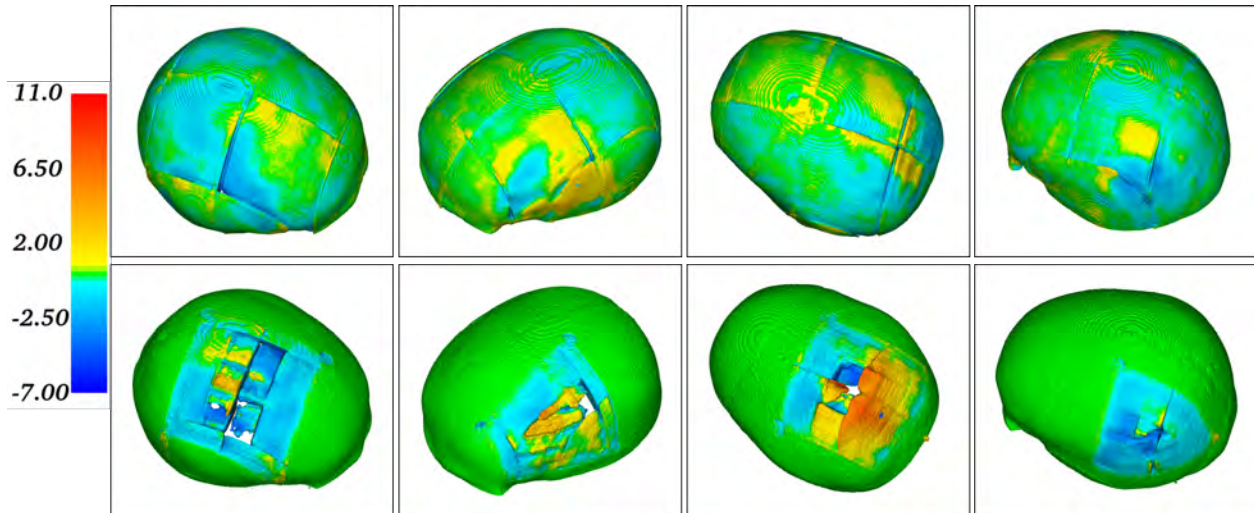
Fig. 6. 3D signed distance colormap between the meshes extracted from the ground truth skulls and the reconstructed skulls by $M_1 - n$ (first row) and $M_2 - n$ (second row). The color ranges for all the colormaps are adjusted to [-7, 11]. Green denotes that the distance is 0. The colormaps are created using 3DMeshMetric (`https://www.nitrc.org/projects/meshmetric3d/`).

for the automatic cranial implant design task. In the middle parts of the implant, the high errors (e.g., yellow or blue) are distributed more sparsely for the ensembled implant than for the implant produced by $M_2 - n$, showing the advantages of implant generation via ensembling. Moreover, we can also see from the distributions of the signed distances (positive/negative) how the curvatures and thickness of the ground truth and the automatically generated implant match.

6. Clinical Applicability

In this section, we discuss the clinical applicability of the proposed automatic implant generation method from the following two aspects: **a)** how the deep learning models trained only on skulls with synthetic defects can also be used for the completion of real-life (craniotomy) defects, **b)** how a cranial implant is assessed for clinical usability and how the quality of the algorithm-produced implants is compared to the present clinical standard. We included the completion result of a craniotomy skull for discussion.

First, as outlined in Section 3, only synthetic skull defects were involved in training the deep learning models. The reasons for not using the craniotomy data directly for training are twofold: **1)** Craniotomy data are much rarer than data of healthy skulls; it is difficult to collect enough craniotomies for training. **2)** Craniotomy cases often lack a pre-operative head CT scan, so that the ground truth of the craniotomy data is unavailable. However, as craniotomy and traumatic defects tend to be much more irregular and complex than the synthetic defects, the deep learning models must be designed and trained to ensure the ability to generalize, which is the key consideration for a deep learning model to function properly in clinical scenarios. To address the challenge, we augmented the skull dataset by creating several random synthetic defects for each healthy skull, and the deep learn models are thereafter trained to complete randomly shaped, positioned and sized defects and consequently craniotomy or traumatic defects.

Second, as our deep learning models are trained to precisely reconstruct a healthy skull prior to the damage (e.g., caused by a brain tumor surgery or head trauma), the resulting implants tend to fit tightly against the boundaries of the defected area, as shown in Figure 5 and Figure 10. However, a tight boundary contact between the implant and the skull is often not necessary and could even be undesirable in cranioplasty procedures, as the defect boundaries might have undergone some remodeling over time through bone regeneration and scarring. We nevertheless considered the tight boundary contact a desirable property, as neurosurgeons can always manually post-process the tightly fitting implants where necessary. In that sense, the goal of our current method is not to automatically generate an implant that is immediately ready for manufacturing and implantation, but to focus on increasing the level of automation for the cranial implant design process using deep learning.

Another positive aspect about the precise reconstruction of the original healthy skull is that a satisfactory cosmetic outcome can be achieved, due to the shape and curvature consistency between the skull and implant, as shown in Figure 5. Besides the implant boundaries, the shape and curvature consistency is of equal importance when assessing the quality of a cranial implant, as one of the primary goals of cranioplasty is to restore a normal skull appearance for the patient.

To evaluate the applicability of the methods to real clinical scenarios, one head CT from a craniotomy case was also processed. In that particular instance, the defect was covered intra-operatively by a titanium mesh for protective purposes. The first column of Figure 9 shows a 3D and axial plane view of the CT scan after segmentation, where the mesh covering the defect is visible. It can be seen that the manually created defect during craniotomy is of greater complexity (e.g., the boundary is not cut straight, and the shape is irregular) than the artificial defects used in training our models, and the existence of the metal mesh on the defected area can also potentially affect the models' performance.
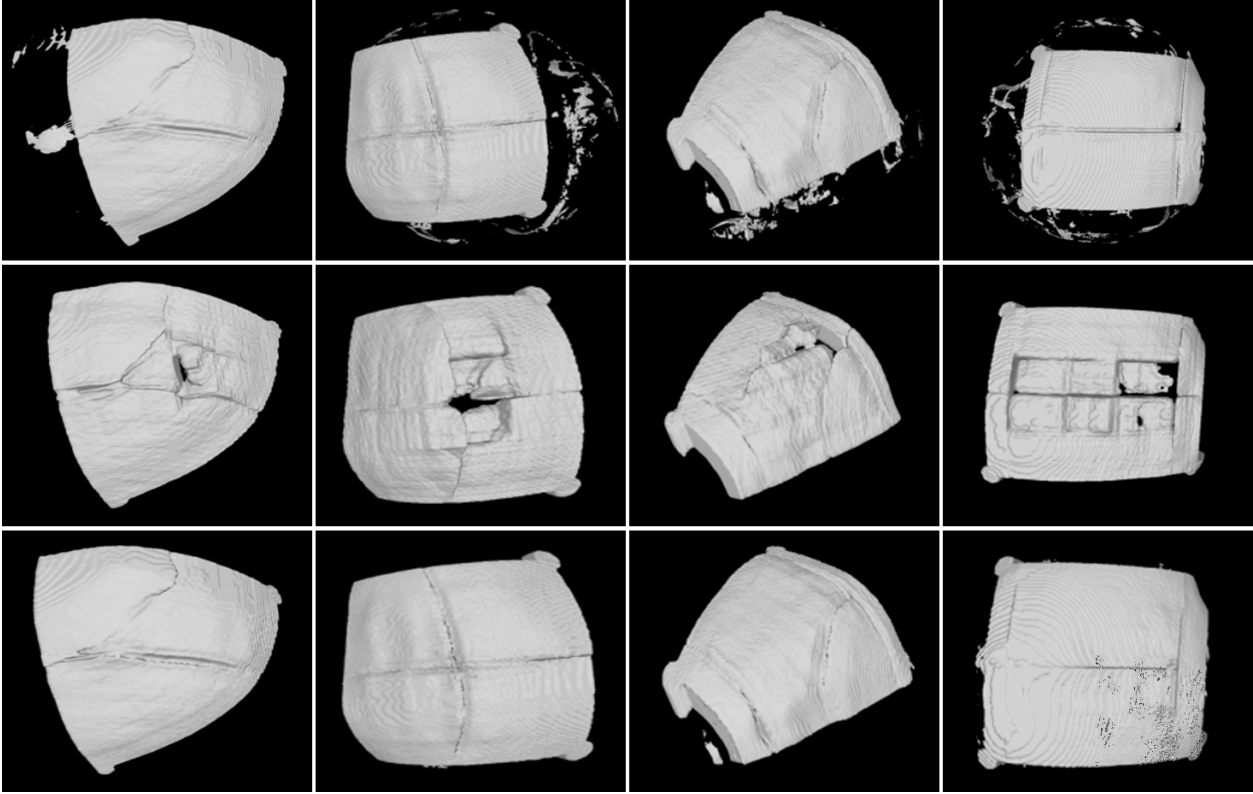
Fig. 7. From first to last row: cranial implants obtained from $M_1 - n$, $M_2 - n$ and the ensembling of $M_1 - n$ and $M_2 - n$, respectively.

Table 3. Mesh distance (measured in millimeters) between the predicted and ground truth implant (pred → gt), the predicted and manually designed implant (pred → manual), the manually designed and ground truth implant (manual → gt).

caseID	pred → gt			pred → manual			manual → gt		
	max	mean	RMS	max	mean	RMS	max	mean	RMS
1	11.3589	0.6995	1.1312	20.8776	0.9422	1.4940	5.8875	1.0419	1.4944
2	5.0173	0.7710	1.0552	7.2673	2.3752	2.9988	6.1991	1.6621	2.2205
3	4.5117	0.6907	0.9956	7.3029	1.9127	2.3855	5.6454	1.5951	2.0507
4	7.0757	0.8466	1.3275	12.0077	1.7596	2.5000	8.6992	1.5536	2.2943

However, the models are still able to perform a satisfactory skull shape completion in such a case, despite the complexity. The second to fourth column of Figure 9 show the completion results produced by $M_1 - n$, $M_1 - o$ and $M_1 - r$, respectively, viewed in 3D and the axial plane. On each 2D plane, the restored areas are highlighted and zoomed in. Based on these limited results, it appears that, even if our models are trained only on healthy skulls with synthetic defects, the models show a promising generalization ability towards real clinical cases with complex defects.

7. Comparison with Commercial Software in Cranial Implant Design

In this section, we compare our fully automatic method (the ensembling of M_1-n and M_2-n) for cranial implant design with a commercial software solution (*Geomagic Studio* https://www.3dsystems.com/software) (van der Meer et al. (2013)).

The cranial implant design workflow using this software is described as follows:

(i) Convert the defective skull volume into a 3D model (.stl), and import the model into Geomagic Studio.

(ii) Make the model watertight using the 'fill' function.

(iii) Trim with curve: select several points alongside the missing region on the model (3 ~ 4 min).

(iv) Fill the missing part using the 'add bridge' and 'fill' functions (6 ~ 8 min).

(v) Boolean operation between the filled model and the original model (4 ~ 6 min).

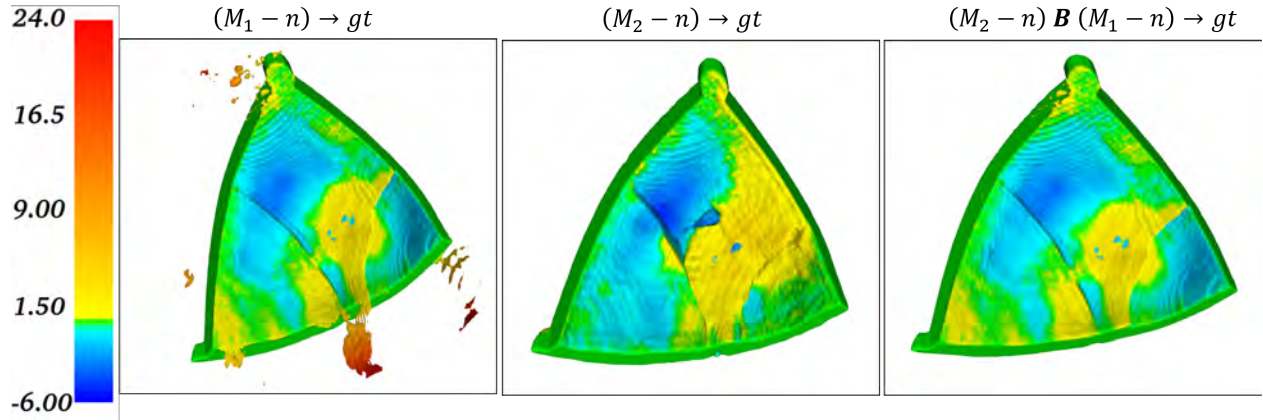(vi) Post-processing: smoothing, denoising, surface curvature adjustment, etc. (18 ~ 25 min).

Fig. 8. 3D signed distance colormap showing the error distributions across the implants obtained from $M_1 - n$ (left), $M_2 - n$ (middle), the ensembling of $M_1 - n$ and $M_2 - n$ (right), respectively. For all the three colormaps, the color ranges are adjusted to [-6, 24], where the color center is set to 0 (green).
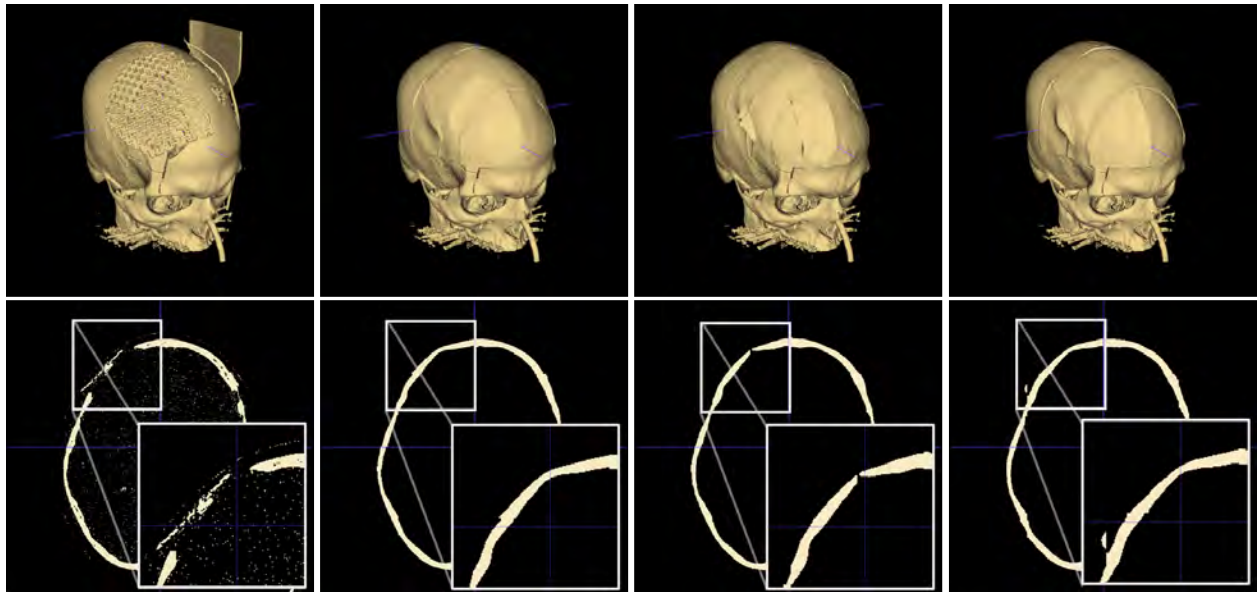

Fig. 9. Skull shape completion on a craniotomy data. The first column shows the craniotomy skull in 3D and axial view. The second to last column shows the skull shape completion results from $M_1 - n$, $M_1 - o$, and $M_1 - r$ respectively.

(vii) Export the implant model as a .stl file.

Designing implants using such modeling software has two principal challenges. First, the shape and curvature of the implant should match that of the surrounding skull to ensure an aesthetic outcome. Second, the thickness and boundary of the implant must fit the defected skull area. The two implant properties need to be manually realized using the software; consequently, the implant quality as well as the design time relies largely on the expertise of the user.

Our method can generate an implant ready for 3D printing in a fully automatic manner in less than one minute, and the implant can fit the hole on the skull with respect to the shape, thickness and borders. In contrast, the design of the implant using commercial software, when performed by a trained user on a standard PC, takes approximately 40 mins, 35 mins, 42 mins, and 45 mins for case 1 ∼ 4 shown in Figure 10.

Figure 10 shows the visual comparison of implant design between our fully automatic method and the software for four cases with large synthetic defects. The second and third row of Figure 10 show the automatically generated (through ensembling) and the manually designed implants, respectively. We can see that the implants obtained from both approaches can fit with the skull defects, while the manually designed implants have smoother surfaces, as the implants are smoothed via a post-processing procedure using the software.

Besides, we show in Figure 11 how the implants produced by the two approaches match with the ground truth implants regarding shape and curvature, from different 2D views. The implants from our algorithm and from the commercial software are shown in red, and the ground truth implants are shown in white. From the 2D views, we can see that the shape and curvature of the implants from both approaches is, to some extent, deviating from the ground truth, while the deviation of the software-produced implant is even larger in the slice shown in Figure 11. Designing an implant that can completely restore
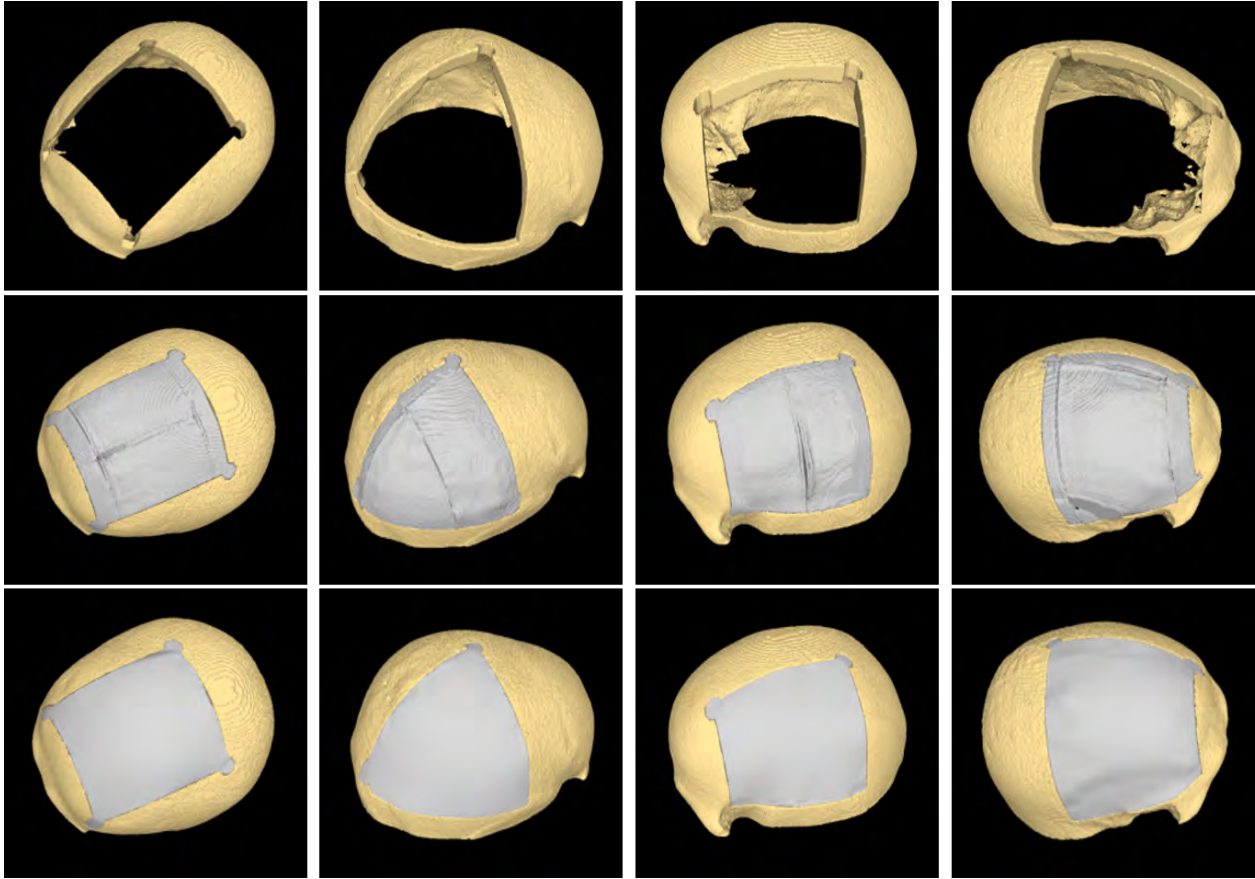
Fig. 10. From the first to last row: the defective skulls, the implants from the proposed approach (ensembling) and the implants from the software. To differentiate, the implants are shown in gray. From left to right, the cases are numbered $1 \sim 4$.

the aesthetics of the original undamaged skull is a difficult task even for experts.

We also quantitatively compared the two approaches. Table 3 shows the mesh distance between the algorithm-produced implant, the software-produced implant and the ground truth implant for the four cases in Figure 10. The mesh distance between two meshes $a_1 \rightarrow a_2$ is measured as follows: For each sampled vertex in a_1, find its closest vertex on the target mesh a_2. Table 3 shows the maximum (max), i.e., the directed Hausdorff distance (HD), the mean and the root mean square (RMS) of the measured closest distances. It should be noted that the software-produced implants are originally in .stl format. To calculate the mesh distances, we extract the meshes from the ground truth and algorithm-produced implants, which are originally voxel grids. Judging from the mean distances and RMS, the automatically produced implants are *closer* to the ground truth than the manually designed implants. The results are understandable, as the user is blind to the original undamaged skulls and therefore the shape and curvature of the implants are adjusted manually based on the user's own experience. In contrast, our deep learning models are trained on a total of $85 \times 9 = 765$ skull pairs to learn the shape distributions of the original healthy skulls. Moreover, it also shows that manual cranial implant design is an ill-posed task, and viable implants produced by different designers do not generally match in terms of quantitative scores. The ground

truth implant obtained by the subtraction of a defective skull from the corresponding original healthy skull is just one of the many possible solutions that our deep learning models are trained to obtain. Furthermore, it should be noted that the original healthy skulls are not always available in clinical situations, but our method can still be used to reconstruct a healthy skull and generate an implant through subtraction.

Table 4 shows DSC, JSC, precision and recall between the algorithm-produced implant, the software-produced implant and the ground truth. The software-produced implants are voxelized for this calculation. Our algorithm beats manual modeling in terms of the quantitative metrics. However, it must be stressed again that the scores only show how similar the implants are to the ground truth, which only represents one of the possible solutions for the cranial implant design task. Similar to the scores reported in Table 1, the scores shown in Table 4 are mediocre even for the manually designed implants (especially for case 2). Mediocre scores do not necessarily indicate low implant quality, due to the ill-posed problem.

8. Discussion

Patch-based training is a widely accepted strategy in deep learning when the data dimensionality is high. In our study, we demonstrate that the conventional patch-based training scheme is suboptimal for shape learning/reconstruction tasks, where the
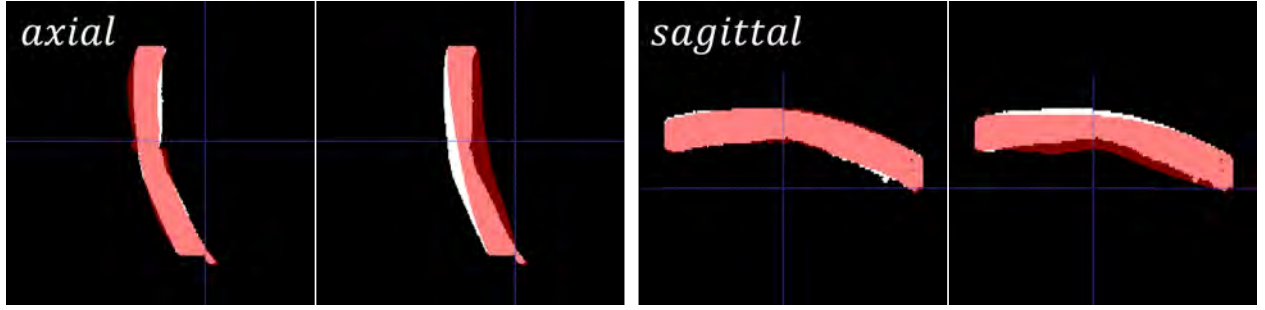
Fig. 11. An overlay of an implant produced by our algorithm and the software (shown in red) onto the ground truth implant (shown in white) from axial and sagittal views. For each view, the implant on the left is from our algorithm and the right is from the software.

Table 4. DSC, JSC, precision and recall between the predicted and ground truth implant (pred ↔ gt), the manually designed and ground truth implant (manual ↔ gt).

caseID	pred ↔ gt				manual ↔ gt			
	DSC	JSC	Precision	Recall	DSC	JSC	Precision	Recall
1	0.8837	0.7916	0.8515	0.9184	0.8802	0.7860	0.8274	0.9401
2	0.8524	0.7428	0.7963	0.9170	0.6446	0.4756	0.6795	0.6131
3	0.8660	0.7637	0.8628	0.8692	0.7924	0.6562	0.7956	0.7893
4	0.8690	0.7684	0.8297	0.9122	0.7951	0.6599	0.7710	0.8209

overall shape characteristics have to be captured, especially when the shapes are of sparse nature, like the skull. By training on randomly cropped patches from the entire training set, the network learns without a prior knowledge of what a particular skull shape is like. The proposed non-overlapping and overlapping training strategies overcome the disadvantage by extracting patches successively from each skull volume.

The training scheme tells the network in an implicit manner, what each specific skull shape is like, which is equivalent to imposing a *virtual* shape constraint or shape prior in the learning process. Increasing the patch size might lead to an improvement in shape learning, as a larger patch carries more information about the overall skull shape, but this is not practical, as the memory resources are limited. Furthermore, if the hole in the skull is much larger than the patch size, the model is expected to fail. For example, when the hole is very large, and a patch cropped from the hole area contains no occupied voxels, the model cannot make a prediction given the empty patch.

The rationale behind using overlapping patches is that reduced patch sparsity can benefit the learning process of deep learning networks, as discussed in Section 4.3 (from the quantitative results in Table 1, M_1-o performs better than M_1-n in skull reconstruction). Increasing the overlapping (current overlapping ratio is 0.5) between two patches can further reduce *relative* patch size, but at the expense of increased computation.

The use of skip connections in an encoder-decoder network for volumetric shape completion tasks can have dual effects, as can be observed from the quantitative and qualitative results. While skip connections have known to be effective in improving model performance for many tasks (e.g., classification), they have presented both beneficial and adverse effects in the skull shape completion task. Using skip connections can weaken

the *hole-filling* ability of the encoder-decoder network and lead to a *non-closed* reconstruction of the implant. The positive role skip connections play in our shape completion task is that they can strengthen the ability to reconstruct the original skull structure. Our proposed model ensembling method can take the advantages of skip connections, while avoiding the adverse effects for implant generation.

9. Conclusion and Future Work

We demonstrated that the medical problem of cranial implant design can be formulated as a high-resolution volumetric shape completion task, which can be fulfilled in a data-driven manner without any explicit shape constraints or priors. Even if we only used healthy skulls with synthetic defects for training the models, our proposed approach shows promise on craniotomy defects. We show how a large skull database can be constructed and then utilized for training deep learning models without the need of human annotation for supervised 3D volumetric shape learning. We propose a patch-based training strategy, which can be beneficial when the data is high resolution and spatially sparse. The proposed training strategy is general and can be used in other applications with similar properties, especially for 3D shape learning tasks. However, one issue associated with the patch-wise training and inference scheme is that, as the contacting borders between two neighboring patches are not always in congruence, merging the patches together to form a complete skull could lead to a bumpy surface, which is undesirable for 3D printing of the skull and implant. Future improvements are needed to address the issue and we will also explore tailored data structures and convolutional operations that can process spatially sparse data more efficiently (Graham

and van der Maaten (2017); Graham et al. (2018)).

Acknowledgment

Appendix A. Generative Adversarial Network

As discussed in Section 2 Related Work, Generative Adversarial Networks (GAN) were also employed in some 3D shape completion studies. In this appendix, we show the skull shape completion results obtained using a GAN and compared the results with those of an auto-encoder (AE) trained using our proposed training strategy $(-n)$. The generator component of the GAN is a standard auto-encoder comprised of four two-strided convolutional layers and four upsampling layers (number of trainable parameters: 22.86 million). The discriminator is a binary classifier consisting of three convolutional layers and one fully connected layer (number of trainable parameters: 5.30 million). For a fair comparison, all the models are trained for 3000 epochs using a mean square error (MSE) loss function, and the auto-encoder used for comparison against the GAN is simply the generator part of the GAN, so that the modules responsible for generating the implants have the same complexity.

The GAN and auto-encoder are trained to predict the implants directly, without the intermediate step of reconstructing the complete skulls. During our experiments, it was found that the GAN (especially the discriminator part) is difficult to converge via a patch-wise training strategy $(-r, -n$ or $-o)$. Therefore, we trained the GAN using the whole but downsampled version of skulls and implants [4]. The auto-encoder is trained on both downsampled data and via a patch based strategy $(-n)$. All the experiments were carried out on a subset of our original dataset described in Section 3. The subset contains only 85 skull-implant pairs for training and 82 pairs for evaluation. The trained model of the GAN and the auto-encoders can be found at `https://github.com/Jianningli/MIA`.
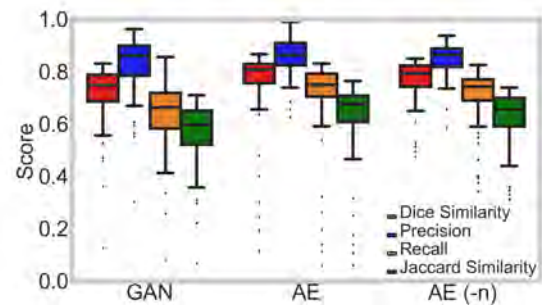
---

[4]The skulls and implants are downsampled to $128 \times 128 \times 64$ and the batch size is set to 4.

Table A.1. Comparison of DSC, JSC, Precision and Recall from GAN and Auto-encoder (AE).

Methods	DSC	JSC	Precision	Recall
GAN	0.7137	0.5646	0.8235	0.6387
AE	0.7589	0.6281	**0.8576**	0.7040
AE $(-n)$	**0.7665**	**0.6282**	0.8489	**0.7045**

Table A.2. *p* values regarding DSC among the trained models.

	DSC (p value)
GAN $\leftrightarrow$ AE	$2.713e^{-2}$
GAN $\leftrightarrow$ AE $(-n)$	$8.650e^{-4}$
AE $\leftrightarrow$ AE $(-n)$	$6.854e^{-1}$

To evaluate the predictions against the ground truth implants, which have a size of $512 \times 512 \times 128$, the predictions from the GAN and AE, when trained on downsampled data, were upsampled to the size of the ground truth. When the AE was trained via a patch based strategy, the output patches were stitched together to form a complete implant of size $512 \times 512 \times 128$. Table A.1 shows the mean scores on the evaluation set. Figure A.1 shows the boxplots of the scores obtained using the three trained models. Table A.2 shows the p values among the trained models regarding DSC. The quantitative results reveal that a standard AE tends to outperform its GAN counterpart in the skull shape completion task by a large margin ($p < 5e^{-2}$, statistically significant). The AE trained using our proposed strategy has the best overall scores. Figure A.2 shows a visual comparison of the predicted implants from the three trained models, from where we can see the advantages of our proposed training method. Due to memory restrictions, the GAN and AE have to be trained on downsampled data. However, downsampling the data from $512 \times 512 \times 128$ to $128 \times 128 \times 64$ can result in severe loss of image quality and learning from the degraded images yields coarse output, as can be seen from the first two columns of Figure A.2.

Fig. A.1. Boxplots of DSC, JSC, precision and recall from the GAN, AE and AE $(-n)$.

In contrast, a patch-wise training and inference strategy can produce high-quality output (the third column of Figure A.2) on the same hardware.
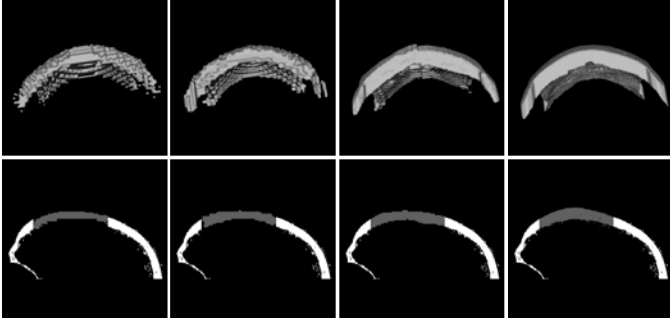
Fig. A.2. A visual comparison of the predicted implants. From the left to right column: the implant produced by GAN, AE, AE ($-n$) and the ground truth. Second row: the predicted implants overlaid onto the defective skull, viewed in sagittal plane.

Table B.1. Comparison of DSC, JSC, Precision and Recall from different loss functions.

Loss	DSC	JSC	Precision	Recall
$\mathcal{L}_{Dice}$	0.7772	0.6483	0.7843	**0.7876**
$\mathcal{L}_{MSE}$	0.7589	0.6281	0.8576	0.7040
$\mathcal{L}_b$	0.6021	0.4413	0.5260	0.7584
$\mathcal{L}_{bdice}$	**0.7880**	**0.6623**	0.8040	0.7868
$\mathcal{L}_{bmse}$	0.7299	0.5947	**0.8599**	0.6624

Appendix B. Boundary Constrained Loss function for Volumetric Shape Completion

In our study, we have discussed the importance of implant boundaries for their clinical utility in cranioplasty and proposed a fully data-driven approach that does not rely on any explicit prior shape information about the implants. In this appendix, we show whether incorporating a shape constraint (more specifically an implant surface/boundary constraint) in the loss function can yield better results compared to using a prior-independent loss function. We used the boundary loss proposed by Kervadec et al. (2021), which was implemented by pixel-wise multiplication of the predicted implants and an Euclidean Distance Transform (EDT) of the ground truth implants.

The base network used here is the same auto-encoder used in Appendix A. For comparison, we train the network on the downsampled subset (images are downsampled to $128 \times 128 \times 64$) using five loss functions i.e., the Dice loss ($\mathcal{L}_{Dice}$), the MSE loss ($\mathcal{L}_{MSE}$), the boundary loss ($\mathcal{L}_b$): $\mathcal{L}_b(y_p, y_g) = \frac{\sum y_p \circ EDT(y_g)}{128 \times 128 \times 64}$, the boundary loss combined with Dice loss ($\mathcal{L}_{bdice}$): $\mathcal{L}_{bdice} = \mathcal{L}_{Dice} + \lambda_{bdice}\mathcal{L}_b$ and the boundary loss combined with MSE loss ($\mathcal{L}_{bmse}$): $\mathcal{L}_{bmse} = \mathcal{L}_{MSE} + \lambda_{bmse}\mathcal{L}_b$. In our experiments, λ_{bdice} and λ_{bmse} were set to 0.01 experimentally. With each loss function, the auto-encoder network is trained for 3000 epochs for direct implant generation, similar to the experiments described in Appendix A. Note that we changed the activation function of the last auto-encoder layer from *tanh* to *sigmoid* when training the network using Dice involved loss functions (i.e., $\mathcal{L}_{Dice}$ and $\mathcal{L}_{bdice}$). The trained models can be found at `https://github.com/Jianningli/MIA`.

The quantitative comparisons are presented at Table B.1 and

Table B.2. *p* values regarding DSC among the trained models.

	DSC (p value)
$\mathcal{L}_{Dice} \leftrightarrow \mathcal{L}_{MSE}$	$3.851e^{-1}$
$\mathcal{L}_{Dice} \leftrightarrow \mathcal{L}_b$	$2.851e^{-15}$
$\mathcal{L}_{Dice} \leftrightarrow \mathcal{L}_{bdice}$	$5.641e^{-1}$
$\mathcal{L}_{Dice} \leftrightarrow \mathcal{L}_{bmse}$	$3.743e^{-2}$
$\mathcal{L}_{bmse} \leftrightarrow \mathcal{L}_b$	$1.617e^{-7}$
$\mathcal{L}_{bmse} \leftrightarrow \mathcal{L}_{bdice}$	$9.623e^{-3}$
$\mathcal{L}_{bmse} \leftrightarrow \mathcal{L}_{MSE}$	$2.320e^{-1}$
$\mathcal{L}_b \leftrightarrow \mathcal{L}_{MSE}$	$2.462e^{-11}$
$\mathcal{L}_{bdice} \leftrightarrow \mathcal{L}_{MSE}$	$1.600e^{-1}$
$\mathcal{L}_{bdice} \leftrightarrow \mathcal{L}_b$	$3.329e^{-17}$

Figure B.1. Table B.2 shows the *p* values among the trained models. It can be seen that, when trained using the boundary loss alone, the AE network tends to have the worst performance. However, combining the boundary loss with Dice loss during training can improve the results compared to using Dice loss or boundary loss alone. Nonetheless, the results seem to suggest that the boundary loss has a negative effect on the MSE loss, even if the influence is not statistically significant ($\mathcal{L}_{bmse} \leftrightarrow \mathcal{L}_{MSE}$). The results also suggest that Dice loss might be more suitable for this skull shape completion task than MSE loss.
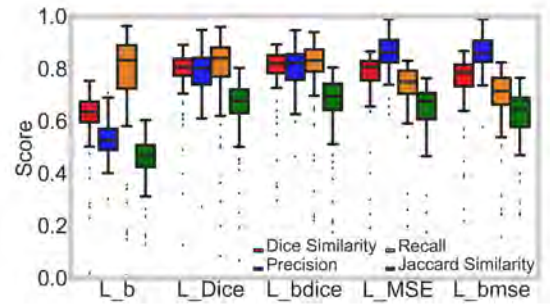

Fig. B.1. Boxplots of DSC, JSC, precision and recall from the AE network trained using $\mathcal{L}_b$, $\mathcal{L}_{Dice}$, $\mathcal{L}_{bdice}$, $\mathcal{L}_{MSE}$, and $\mathcal{L}_{bmse}$.

A visual comparison of the predicted implants is shown in Figure B.2. From the 2D views, it can be seen that training using the boundary loss ($\mathcal{L}_b$) alone tends to result in over-segmentation, and the boundary loss is less successful than the Dice and MSE loss in preserving fine details, such as the small roundish corners of the implants. In contrast, Dice and MSE loss tend to lead to under-segmentation. More experiments are still needed to decide an optimal coefficient (λ_{bdice} and λ_{bmse}) for the hybrid loss function ($\mathcal{L}_{bdice}$ and $\mathcal{L}_{bdice}$).

Appendix C. 3D Printed Cranial Implants

A surface model (triangular mesh) has to be extracted from the algorithm's output (binary voxel grid) in order to be 3D printed. Figure C.1 (A, B) show a 3D printed implant and the corresponding defective skull (cropped). It was found
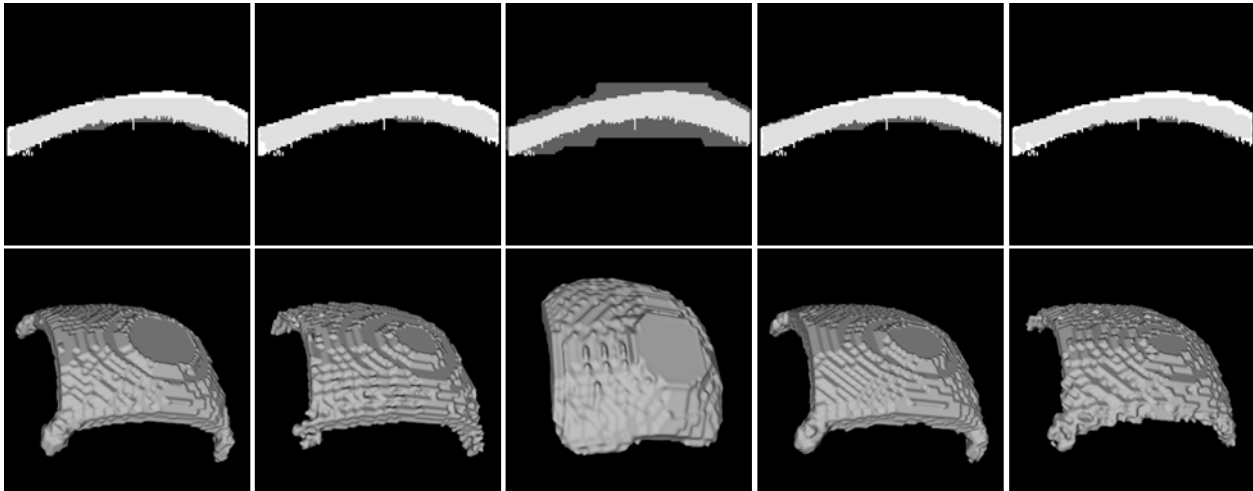
Fig. B.2. From left to right: an overlay of the implants from the network trained using $\mathcal{L}_{Dice}$, $\mathcal{L}_{MSE}$, $\mathcal{L}_b$, $\mathcal{L}_{bdice}$ and $\mathcal{L}_{bmse}$ (gray) onto the ground truth (while). Second row: the corresponding implants viewed in 3D.

that, even if the implant can fit the skull perfectly in software (e.g., Figure 5 and Figure 10 in the main manuscript), there is problem inserting the 3D printed implant into the printed skull. This is due to the fact that the implant is printed using non-elastic material and has to be manually rasped (around the borders) so that it is able to be inserted into the skull. In cranioplasty, commercial implants sometimes also require manual rasping before used in the surgeries. Figure C.1 (C) shows an implant produced by our algorithm placed on top of the corresponding manually designed implant.

Fig. C.1. (A) and (B): A 3D printed implant and the corresponding defective skull. (C): Implant designed automatically by the proposed method (ensembling) placed on top of a manually designed implant.

References

Angelo, L., Di Stefano, P., Governi, L., Marzola, A., Volpe, Y., 2019. A robust and automatic method for the best symmetry plane detection of craniofacial skeletons. Symmetry 11, 245. doi:10.3390/sym11020245.

Bose, S., Traxel, K.D., Vu, A.A., Bandyopadhyay, A., 2019. Clinical significance of three-dimensional printed biomaterials and biomedical devices. MRS bulletin 44(6), 494–504.

von Campe, G., Pistracher, K., 2020. Patient specific implants (psi), in: Cranial Implant Design Challenge, Springer. pp. 1–9.

Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al., 2015. Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 .

Chen, X., Xu, L., Li, X., Egger, J., 2017. Computer-aided implant design for the restoration of cranial defects, in: Scientific Reports, pp. 1–10.

Chia, H.N., Wu, B.M., 2015. Recent advances in 3d printing of biomaterials. Journal of Biological Engineering 9.

Dai, A., Qi, C.R., Nießner, M., 2016. Shape completion using 3d-encoder-predictor cnns and shape synthesis. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 6545–6554.

Dou, Q., Yu, L., Chen, H., Jin, Y., Yang, X., Qin, J., Heng, P.A., 2017. 3d deeply supervised network for automated segmentation of volumetric medical images. Medical Image Analysis 41, 40–54.

Egger, J., Gall, M., Tax, A., Üçal, M., Zefferer, U., Li, X., von Campe, G., Schäfer, U., Schmalstieg, D., Chen, X., 2017. Interactive reconstructions of cranial 3d implants under mevislab as an alternative to commercial planning software. PLoS ONE 12, 20. doi:10.1371/journal.pone.0172694.

Friebe, M., Sanchez, J., Balakrishnan, S., Illanes, A., Nagaraj, Y., Odenbach, R., Matooq, M., Krombach, G.A., Vogele, M., Boese, A., 2018. In-room ultrasound fusion combined with fully compatible 3d-printed holding arm – rethinking interventional mri. Medical Devices (Auckland, N.Z.) 11, 77 – 85.

Fuessinger, M.A., Schwarz, S., Cornelius, C.P., Metzger, M.C., Ellis, E., Probst, F., Semper-Hogg, W., Gass, M., Schlager, S., 2017. Planning of skull reconstruction based on a statistical shape model combined with geometric morphometrics. International Journal of Computer Assisted Radiology and Surgery 13, 519–529.

Gall, M., Li, X., Chen, X., Schmalstieg, D., Egger, J., 2016. Computer-aided planning and reconstruction of cranial 3d implants. 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) , 1179–1183.

Gall, M., Tax, A., Li, X., Chen, X., Schmalstieg, D., Schäfer, U., von Campe, G., Egger, J., 2019. Cranial Defect Datasets. Figshare URL: https://figshare.com/articles/Cranial_Defect_Datasets/4659565, doi:10.6084/m9.figshare.4659565.v6.

Graham, B., Engelcke, M., Van Der Maaten, L., 2018. 3d semantic segmentation with submanifold sparse convolutional networks, in:
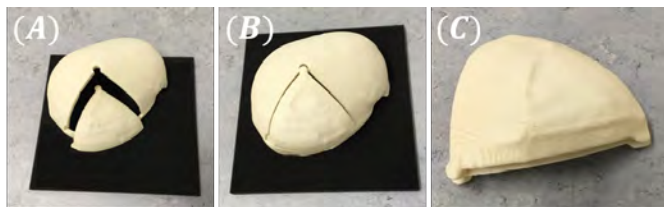
Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 9224–9232.

Graham, B., van der Maaten, L., 2017. Submanifold sparse convolutional networks. arXiv preprint arXiv:1706.01307 .

Han, X., Li, Z., Haibin, H., Kalogerakis, E., Yu, Y., 2017. High-resolution shape completion using deep neural networks for global structure and local geometry inference, pp. 85–93. doi:10.1109/ICCV.2017.19.

Heinrich, M.P., Oktay, O., Bouteldja, N., 2019. Obelisk-net: Fewer layers to solve 3d multi-organ segmentation with sparse deformable convolutions. Medical Image Analysis 54, 1–9.

Kamnitsas, K., Ledig, C., Newcombe, V.F.J., Simpson, J.P., Kane, A.D., Menon, D.K., Rueckert, D., Glocker, B., 2017. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. Medical Image Analysis 36, 61–78.

Kazhdan, M.M., Bolitho, M., Hoppe, H., 2006. Poisson surface reconstruction, in: Symposium on Geometry Processing, pp. 1–13.

Kazhdan, M.M., Hoppe, H., 2013. Screened poisson surface reconstruction. ACM Trans. Graph. 32, 29:1–29:13.

Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., Ayed, I.B., 2021. Boundary loss for highly unbalanced segmentation. Medical image analysis 67, 101851.

Kung, W.M., Chen, S.T., Lin, C.H., Lu, Y.M., hsuan Chen, T., Lin, M.S., 2013. Verifying three-dimensional skull model reconstruction using cranial index of symmetry, in: PloS one. doi:https://doi.org/10.1371/journal.pone.0074267.

Li, D., Shao, T., Wu, H., Zhou, K., 2017. Shape completion from a single rgbd image. IEEE Transactions on Visualization and Computer Graphics 23, 1809–1822.

Li, J., Chen, B.M., Lee, G.H., 2018a. So-net: Self-organizing network for point cloud analysis. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition , 9397–9406.

Li, X., Dou, Q., Chen, H., Fu, C.W., Qi, X., Belavý, D.L., Armbrecht, G., Felsenberg, D., Zheng, G., Heng, P.A., 2018b. 3d multi-scale fcn with random modality voxel dropout learning for intervertebral disc localization and segmentation from multi-modality mr images. Medical Image Analysis 45, 41–54.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018c. Pointcnn: Convolution on x-transformed points, in: NeurIPS, pp. 820–830.

Litany, O., Bronstein, A.M., Bronstein, M.M., Makadia, A., 2017. Deformable shape completion with graph convolutional autoencoders. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition , 1886–1895.

Liu, X., Yan, M., Bohg, J., 2019. Meteornet: Deep learning on dynamic 3d point cloud sequences. ArXiv abs/1910.09165.

Marzola, A., Governi, L., Genitori, L., Mussa, F., Volpe, Y., Furferi, R., 2019. A semi-automatic hybrid approach for defective skulls reconstruction. Computer-Aided Design and Applications 17, 190–204. doi:10.14733/cadaps.2020.190-204.

van der Meer, W.J., Bos, R.R., Vissink, A., Visser, A., 2013. Digital planning of cranial implants. The British journal of oral & maxillofacial surgery 51, 450–452. doi:https://doi.org/10.1016/j.bjoms.2012.11.012.

Mitra, N.J., Guibas, L.J., Pauly, M., 2006. Partial and approximate symmetry detection for 3d geometry. ACM Trans. Graph. 25, 560–568.

Morais, A., Egger, J., Alves, V., 2019. Automated Computer-aided Design of Cranial Implants Using a Deep Volumetric Convolutional Denoising Autoencoder. pp. 151–160.

Ngo, H.T.M., Lee, W.S., 2011. Feature-first hole filling strategy for 3d meshes, in: ICCV 2011.

Park, J.M., Son, J., An, H.J., Kim, J.H., Wu, H.G., Kim, J.I., 2019. Bio-compatible patient-specific elastic bolus for clinical implementation. Physics in medicine and biology 64(10), 105006.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2016. Pointnet: Deep learning on point sets for 3d classification and segmentation. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 77–85.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: NIPS.

Rengier, F., Mehndiratta, A., von Tengg-Kobligk, H., Zechmann, C.M., Unterhinninghofen, R., Kauczor, H.U., Giesel, F.L., 2010. 3d printing based on imaging data: review of medical applications. International Journal of Computer Assisted Radiology and Surgery 5, 335–341.

Sakr, N.M., Youssef, B.B.A., Hassan, Y.F., Atta, E.H., 2018. An effective method for hole filling in 3d triangular meshes. 2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT) ,

1–7.

Sarmad, M., Lee, H.J., Kim, Y.M., 2019. Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5898–5907.

Schiebener, D., Schmidt, A., Vahrenkamp, N., Asfour, T., 2016. Heuristic 3d object shape completion based on symmetry and scene context. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) , 74–81.

Stutz, D., Geiger, A., 2018. Learning 3d shape completion from laser scan data with weak supervision. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition , 1955–1964.

Sung, M., Kim, V.G., Angst, R., Guibas, L.J., 2015. Data-driven structural priors for shape completion. ACM Trans. Graph. 34, 175:1–175:11.

Wang, J., Noble, J.H., Dawant, B.M., 2019. Metal artifact reduction for the segmentation of the intra cochlear anatomy in ct images of the ear with 3d-conditional gans. Medical image analysis 58, 101553.

Wang, W., Huang, Q., You, S., Yang, C., Neumann, U., 2017. Shape inpainting using 3d generative adversarial network and recurrent convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 2298–2306.

Wu, R., Chen, X., Zhuang, Y., Chen, B., 2020. Multimodal shape completion via conditional generative adversarial networks, in: The European Conference on Computer Vision (ECCV).

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2014. 3d shapenets: A deep representation for volumetric shapes. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 1912–1920.

Yang, X., Bian, C., Yu, L., Ni, D., Heng, P.A., 2017a. Hybrid loss guided convolutional networks for whole heart parsing, in: Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS ChallengesI, pp. 215–223.

Yang, Y., Feng, C., Shen, Y., Tian, D., 2017b. Foldingnet: Point cloud auto-encoder via deep grid deformation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition , 206–215.

Zhao, W., Gao, S., Lin, H., 2007. A robust hole-filling algorithm for triangular mesh, pp. 22–22. doi:10.1109/CADCG.2007.4407836.