Jianning Li, BSc

Deep Learning for Cranial Defect Reconstruction

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Computer Science

submitted to

Graz University of Technology

Supervisors

Priv.-Doz. Dr. Dr. Jan Egger
Univ.-Prof. Dipl-Ing. Dr.techn. Dieter Schmalstieg

Institute for Computer Graphics and Vision
Head: Univ.-Prof. Dipl-Ing. Dr.techn. Dieter Schmalstieg

Graz, Austria, January 2020

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date	Signature

Acknowledgement

This work was supported by CAMed (COMET K-Project 871132), which is funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT), and the Austrian Federal Ministry for Digital and Economic Affairs (BMDW), and the Styrian Business Promotion Agency (SFG).

In cooperation with

Abstract (English)

A fast and fully automatic design of 3D printed patient-specific cranial implants is highly desired in cranioplasty, a 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 this end, a deep neural network based on the encoder-decoder architecture is adopted. To facilitate supervised training, a database containing 167 healthy skulls segmented from head CT in clinical routine has been established. Synthetic defects are injected into the healthy skull to create training and evaluation data pairs. To work on high-resolution and spatially sparse skull data, we proposed a tailored patch-based training scheme that overcomes the disadvantages of conventional patch-based training method and shows significant improvement. In particular, the patch-based training method is applied to images of high resolution and proves to be effective in tasks such as segmentation. However, we demonstrate that conventional patch-based training method is suboptimal for tasks such as shape reconstruction, where the overall shape distribution of the target has to be learnt, since if cannot be captured efficiently by a sub-volume cropped from the target. Additionally, the standard dense implementation of a convolutional neural network (CNN) tends to performs poorly on sparse data such as the skull, which has a low voxel occupancy rate. Our tailored training scheme encourages a standard CNN to learn interpretable features from the high-resolution and sparse data. We have evaluated our method on both skulls with synthetic defects and skulls with real defects manually injected by neurosurgeons in craniotomy, and the results show potential for clinical applicability.

Abstract (German)

Bei der Kranioplastik (Prozess zur Wiederherstellung nach einem Schädeldefekt) ist eine schnelle und vollautomatische Konstruktion von 3D-gedruckten patientenspezifischen Schädelimplantaten sehr wünschenswert. Wir formulieren die Wiederherstellung nach Schädeldefekten als eine volumetrische Vervollständigung der Schädelform, bei der ein Teil des Schädelvolumens (der Schädeldefekt) automatisch vervollständigt wird. Der Unterschied bzw. die Differenz zwischen dem fertigen (wiederhergestellten) Schädel und dem Schädel mit Defekt ist also der wiederhergestellte Defekt, d.h., das Implantat, das für die Kranioplastik verwendet werden kann. Zu diesem Zweck wird ein tiefes neuronales Netz verwendet, das auf der Encoder-Decoder-Architektur basiert. Zur Erleichterung des sogenannten beaufsichtigten Trainings wurde eine Datenbank mit 167 gesunden Schädeln aufgebaut, die in der klinischen Routine akquiriert wurden und bei denen der Schädelknochen segmentiert wurde. Um Trainings- und Bewertungsdatenpaare für das neuronale Netz zu erstellen, wurden synthetische Defekte in die gesunden Schädel injiziert. Um mit den hochauflösenden und räumlich spärlichen Schädeldaten arbeiten zu können, haben wir ein maßgeschneidertes patch-basiertes Trainingsschema vorgeschlagen, das die Nachteile herkömmlicher patch-basierter Trainingsmethoden überwindet und signifikante Verbesserungen aufweist. Insbesondere wird eine patch-basierte Trainingsmethode angewendet, wenn das gesamte Bild eine hohe Auflösung aufweist. Dies ist bei Aufgaben wie der Segmentierung effektiv. Wir zeigen jedoch, dass herkömmliche, auf Patches basierende Trainingsmethoden für Aufgaben wie die Formrekonstruktion nicht optimal sind, weil die Gesamtformverteilung des Ziels (die von einem Teilvolumen nicht effizient erfasst werden kann) gelernt werden muss. Zusätzlich neigt die standardmäßig dichte Implementierung eines Convolutional Neural Network (CNN) dazu, bei spärlichen Daten schlecht zu arbeiten. Leider sind

die Schädeldaten bei einer geringen Voxelbelegungsrate räumlich spärlich. Unser maßgeschneidertes Trainingsschema erlaubt es einem Standard-CNN, aus den hochauflösenden und spärlichen Daten Features zu lernen. Wir haben unsere Methode sowohl an Schädeln mit synthetischen Defekten als auch an Schädeln mit echten Defekten, die von Neurochirurgen bei der Kraniotomie manuell injiziert wurden, evaluiert. Die Ergebnisse zeigen, dass eine klinische Anwendung potenziell möglich ist.

Contents

List of Figures

1 Introduction

1.1 Medical Background

Cranioplasty is the surgical process where the skull defect, caused in a brain tumor surgery or by trauma, is repaired using a cranial implant, which must fit precisely against the borders of the skull defect to replace the removed cranial bone. The designing of the cranial implant is a challenging task and involves several steps: (1) obtaining the 3D imaging data of the skull with the defect from computed tomography (CT) or magnetic resonance imaging (MRI) scans, (2) converting the 3D imaging data into 3D mesh models, and (3) creating the 3D model of the implant for additive manufacturing aka 3D printing. The last step usually requires expensive commercial software, which clinical institutions often have limited access to. Researchers have been working on CAD software as alternative to the commercial software for the designing of cranial implant, but these approaches still involve human interaction. All these software tools are time-consuming and require expertise of the specific medical domain. Therefore, a fast and fully automatic design of cranial implants is highly desired in cranioplasty, which also enables intra-operative 3D-printing of the implants for the patient for an instant defect reconstruction in one intervention/setting. This course of action stands in strong contrast to the current clinical routine, where the implant is manufactured offline, most often by an external manufacturer. This means that the patient has to undergo surgery after some days or even weeks, when the implant has been manufactured, including repeated anesthesia. Furthermore, during the time the implant is manufactured, the patient retains the cranial defect and lacks sufficient protection of the brain in the defected area.

1.2 Scientific Contributions

The contribution of this thesis lies in five aspects. *First*, we demonstrated the feasibility of using a 3D encoder-decoder network for volumetric shape completion and successfully applied the network to fully automatic skull defect restoration and cranial implant generation. *Second*, we constructed a large database of healthy (complete) skulls and showed how to inject synthetic defects into the healthy skulls in order to facilitate the training of the deep learning networks. The database can be used for various other purposes (e.g., creating 3D skull atlas and skull anatomy analysis) and is scheduled to be open-sourced shortly. *Third*, we proposed a tailored patch-based strategy for training deep learning networks when the data are spatially sparse and high resolution (e.g., $512 \times 512 \times Z$), which shows a significant improvement compared with using a conventional training scheme. *Fourth*, we established an interpretable deep learning model based on the encoder-decoder architecture for skull shape completion, which can serve as the baseline for future work that makes use of our datasets. *Fifth*, our approach has been evaluated on skulls with real defect from craniotomy and the results show promise for clinical applicability.

1.3 Thesis Outline

My thesis is organized as follows: *Chapter* 2 reviews the existent semi-automatic approaches for cranial implant design and introduces the concept and commonly used methods for 3D shape completion in computer graphics, where the data to be processed are usually triangular meshes or 3D point clouds. We further extend the concept of 3D shape completion to high-resolution volumetric data, i.e., the skull. *Chapter* 3 introduces the skull database we constructed and the methods we used for pre-processing and synthetic defect injection. *Chapter* 4 gives the details of the methodology including the network architecture and the training algorithms. *Chapter* 5 presents the experiments that extensively evaluated the proposed methods and the discussion of the results. *Chapter* 6 is the discussion of how the proposed training strategy overcomes the disadvantage of conventional

random cropping and benefits deep learning when the training data are sparse and of high resolution. Finally, *Chapter* 7 concludes the study and outlines areas for future research.

2 Related Work

There exist several semi-automatic CAD approaches for skull defect restoration and cranial implant reconstruction, which exploit the symmetricity of the human skulls and fills the missing data by mirroring (Gall, Xing Li, et al., 2016; X. Chen et al., 2017; Marzola et al., 2019; Egger et al., 2017). These approaches are not optimal, considering that human skulls are not strictly symmetric. Moreover, these approaches are still time-consuming, limiting the application for intra-operative implant reconstruction and an instant manufacturing of the implant with a bio-compatible 3D printer. In (Morais, Egger, and Alves, 2019) and (Morais, 2018), a deep learning approach based on volumetric auto-encoder for fully automatic skull defect restoration was proposed. This approach is limited to a low-resolution volumetric grid (30^3, 60^3 and 120^3) generated from MRI data. Skull defect restoration can be formulated as a 3D volumetric shape completion task, aiming at predicting the missing structure 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, Bolitho, and Hoppe, 2006; Kazhdan and Hoppe, 2013; Zhao, Gao, and Lin, 2007; Ngo and W.-S. Lee, 2011; Sakr et al., 2018). Some approaches complete the shape by exploiting the symmetricity of 3D shapes represented as point cloud (Schiebener et al., 2016; Sung et al., 2015; Mitra, Guibas, and Pauly, 2006). The exploitation of symmetricity for shape completion is similar to some of the interactive approaches for skull defect restoration, which utilize the symmetricity of the human skull (Angelo et al., 2019; Marzola et al., 2019; Gall, Xing Li, et al., 2016; Egger et al., 2017; X. 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 datasets. These approaches usually operate on a volumetric representation of a 3D point cloud from 3D scanning or RGBD images, such as (truncated) signed

distance field (TSDF), using traditional 3D convolutional neural networks, which are mostly based on an encoder-decoder architecture (Dai, Qi, and Nießner, 2016; Stutz and Geiger, 2018; Han et al., 2017; D. Li et al., 2017; Litany et al., 2017). The volumetric representation of a point cloud is usually binary, with 1 representing that the corresponding voxel grid is occupied by a point, and 0 representing that the voxel gird is not occupied. However, these approaches specifically target the shape completion of 3D point clouds or mesh data structures. Even if some deep learning approaches utilize the volumetric representation of point clouds, they are limited to process low-resolution volume such as 128^3 or 256^3. For the restoration of a cranial defect, or, in other words, for skull shape completion, the original data structure is volumetric and requires high-resolution input and output of the dimension $512^2 \times Z$ (in our case, Z ranges from 255 to 480). Recent advances in deep learning have enabled convolutional operations to be applied directly to point cloud data structures. Point cloud based deep learning approaches for 3D shape classification and segmentation often use the encoder-decoder architecture (Qi, Su, et al., 2016; Qi, Yi, et al., 2017; J. Li, B. M. Chen, and G. H. Lee, 2018; Y. Yang et al., 2017; Y. Li et al., 2018; Liu, Yan, and Bohg, 2019). However, most of the proposed approaches are targeted at small point clouds (Wu et al., 2014). The lossless conversion of a single skull volume to its corresponding 3D surface model yields a large mesh with, e.g., 1.4 million points and 2.8 million triangular faces, which these existing approaches are unable to handle. In this work, we propose a data-driven approach for volumetric shape completion, dealing directly with high-resolution volumetric data. The results show that the proposed approach can effectively restore the fine details of the missing cranial structures, while maintaining the original structure of the defected skull.

3 Materials

3.1 Datasets

We constructed a database containing 167 healthy (complete) skull datasets. The skulls are segmented manually from CT scans of the head during the clinical routine by neurosurgeons and have a high resolution of $512 \times 512 \times Z$ (Z ranges from 255 to 480). Each dataset has the complete anatomical structure of a skull without a defect. To create training pairs out of the complete skull data, we empirically injected defects into the complete skulls to simulate the defect of a craniotomy. Afterwards, the data driven approach will learn to refill the defect based on these data pairs. This course of action has two main advantages: (1) By injecting the holes into healthy skulls, we know the ground truth for a reconstruction. (2) We can inject several different holes for each healthy skull; Hence, we have more data for training and evaluation than the actual number of skulls, which is a kind of data augmentation.

3.2 Preprocessing and Data Pair Creation

Figure 3.1 shows the skull segmentation from CT images. After segmentation, the CT table and the noise inside the skull are removed using 3D connected component analysis. Depending on the age, neurosurgeons use a customized segmentation threshold $(100 - 200)$ for each skull to make sure the complete skull anatomy (including the maxillary sinus, which is very thin) can be preserved after thresholding. Besides the skull, the process can also preserve structures with similar density to the skull bones, such as calcium, which is the cause of noise inside the skull.

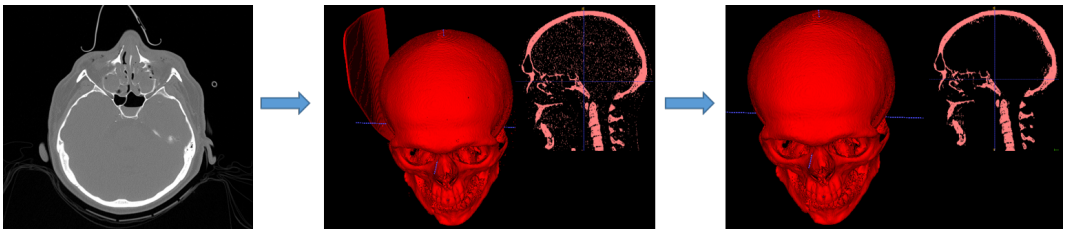

Figure 3.1: Skull segmentation from CT images and denoising using 3D connected component analysis.

In order to create realistic defects in the healthy skull, we refer to the real morphology of skull defects that are manually injected by neurosurgeons during craniotomy. Figure 3.2 (a) shows a defected skull acquired in craniotomy from our previously published dataset (Gall, Tax, et al., 2019). In craniotomy, the surgeons open the skull manually using craniotome by drilling roundish holes in the corners. The drill comes in different diameters, and 13mm/9mm (outer/inner diameter) is commonly used. Figure 3.2 (b) shows the synthetic defects we injected into the healthy skull after noise reduction. For each single skull, we considered several defects with different position and size to enlarge the datasets for training and evaluation. The defects used in the datasets are simplified, but represent the general morphology of real defects (the real defect has a rougher border). As is shown in Figure 3.2 (b), we injected nine random defects into each skull, resulting in a total of $167 \times 9 = 1503$ data pairs (nine defected skulls correspond to one healthy skull). The data pairs are further split into a training set of 765 (85×9) pairs and a testing set of 738 (82×9) pairs. Besides the 167 healthy skulls, we also collected two additional CT scans from craniotomy for the evaluation of the approach.

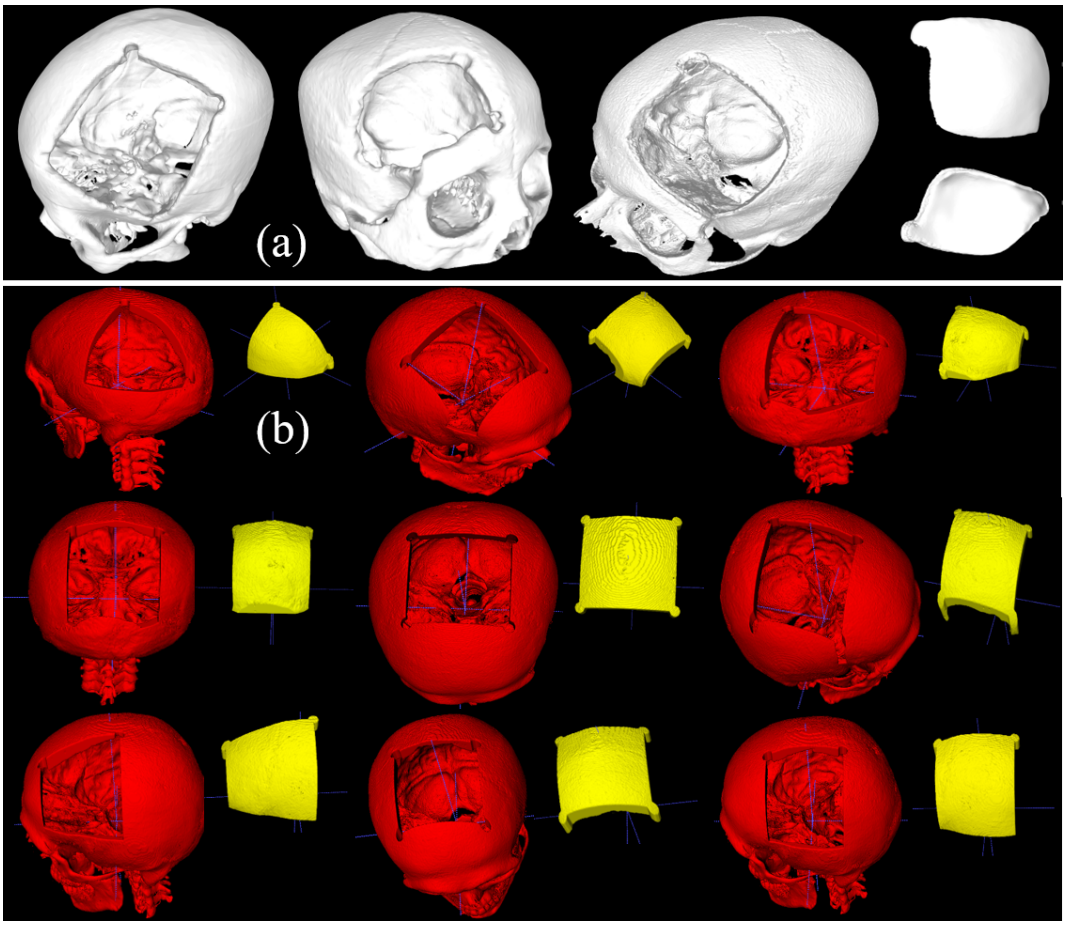

Figure 3.2: Illustration of the skull defect in craniotomy (a) and the simplified defect (b) used in our datasets. We considered nine random defects for each healthy skull.

4 Methodology

4.1 Probabilistic Model for Volumetric Shape Completion

Restoring defects in the skull can be formulated as a 3D volumetric shape completion task, where the skull is represented as a 3D volume with a missing part (see Figure 3.2 (b)). Let $S_d \in [0,1]$ be the skull with a defect, and, $S_c \in [0,1]$ be the corresponding complete skull of the same dimension $R^{512\times512\times Z}$, where 1 represents voxels belonging to the skull and 0 represents unoccupied voxels (the background and the empty space inside the skull) in the skull volume. The skull defect restoration is to reconstruct S_c from $S_d : S_d \overset{\phi(\theta)}{\rightarrow} S_c$, $\phi(\theta)$ is the reconstruction matrix with parameters θ, such that their difference $L(S_c, \phi(\theta)S_d) = \{S_c - \phi(\theta)S_d\}$ can be minimized. To make the optimization more stable and compensate over-fitting, $L2$ kernel (weights) regularizers are introduced:

$$\min_{\theta} L(S_c, \phi(\theta)S_d) + \lambda \|\theta\|_2^2, \lambda = 0.0005 \tag{4.1}$$

The restored defect or, in other words, the cranial implant I_{mp} can be expressed as the subtraction of S_d from the reconstructed skull $\phi(\theta)S_c$:

$$I_{mp} = \phi(\theta)S_d - S_d \tag{4.2}$$

Similarly, the ground truth implant can be obtained by subtracting S_d from S_c: $I_{mg} = S_c - S_d$. We demonstrated that minimizing the difference between I_{mg} and I_{mp} is equivalent to minimizing the difference between S_c and $\phi(\theta)S_d$, as in Equation (4.1): $I_{mg} - I_{mp} = (S_c - S_d) - (\phi(\theta)S_d - S_d) = S_c - \phi(\theta)S_d$. We consider a parameterized probabilistic model $P(S_c|S_d, \theta)$

with parameters θ that maps a partial skull S_d to a complete skull S_c, which are both represented as a 3D volume. The model serves as the reconstruction matrix $\phi(\theta)$ and aims at solving Equation (4.1).

The output of the model is given as $p = P(S_c|S_d, \theta)$, where $p \sim (0, 1)$ is the probability of the voxel being occupied in S_c given the input S_d. From a probabilistic perspective, solving Equation (4.1) is equivalent to directly maximizing the conditional probability p and the solution θ^* can be obtained by:

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, P(S_c|S_d, \theta) \tag{4.3}$$

Considering that S_d , S_c and p are of the same volumetric dimension, it is natural to think of using a 3D encoder-decoder network for modeling $p = P(S_c|S_d, \theta)$, where the number of operations for down-sampling and up-sampling is equal, and the output is the probability of the voxel being occupied. We use the Dice Similarity Coefficient (DSC) as the loss function, as shown in Equation (4.4). The negative value of the DSC between $\phi(\theta)S_d$ and S_c is used to guide the updating of the model parameter θ during optimization (considering (S_d, S_c) as a training pair):

$$L(S_c, p(\theta, S_d)) = \frac{-2p(S_c, \theta)S_c}{p(S_c, \theta)p(S_c, \theta) + S_c S_c} + \lambda \|\theta\|_2^2, \lambda = 0.0005 \tag{4.4}$$

4.2 Dimensionality Problem in Deep Learning for Medical Image Processing

Medical images are usually of high dimensionality (e.g., $512 \times 512 \times Z$ for our skull data) and cannot be fed into deep learning networks directly due to the limitation of GPU memory. We rejected down-sampling the images to a lower resolution as it causes loss of information and deforms the anatomical structure. Instead, we consider a patch-based solution for the dimensionality problem: we crop a smaller sub-volume from the high-resolution image as the input to the deep learning network (X. Yang et al., 2017; Wang, Noble,

and Dawant, 2019; Heinrich, Oktay, and Bouteldja, 2019; Dou et al., 2017; Xiaomeng Li et al., 2018; Kamnitsas et al., 2017).

4.3 Network Architecture

Considering the common positions of defects in craniotomy, we use only the upper part of the skull for network training and testing. The upper part of the skull is cropped from each skull, resulting in a fixed skull dimension of $512 \times 512 \times 128$ for each dataset. We constructed two baseline encoder-decoder networks, which are illustrated in Figure 4.1 and Figure 4.2. For the network in Figure 4.1, the encoder is comprised of four convolutional layers with a stride of two for down-sampling and an additional convolutional layer with a stride of one for feature embedding. The decoder is comprised of four deconvolutional layers for up-sampling, and the output layer is a convolutional layer with a stride of one each. The latent feature space and the output layer are represented by convolutional layers with a stride of one. The total number of trainable parameters of the model is 82.076 million. In Figure 4.2, 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 this network is 41.024 million.

4.4 Training Strategy

We propose two patch-based strategies for training the auto-encoder in Figure 4.1 and Figure 4.2: ***overlapping cropping*** and ***non-overlapping cropping*** for the comparison with the conventional ***random cropping***. For each training strategy, we consider a fixed patch dimension of 128^3 as the input of the network. For convenience of notation, we define the following:

The training set

$$Tr\left\{\left\{S_{d1}, S_{g1}\right\}, \left\{S_{d2}, S_{g2}\right\} \ldots \left\{S_{di}, S_{gi}\right\} \ldots \left\{S_{dN_{tr}}, S_{gN_{tr}}\right\}\right\}$$

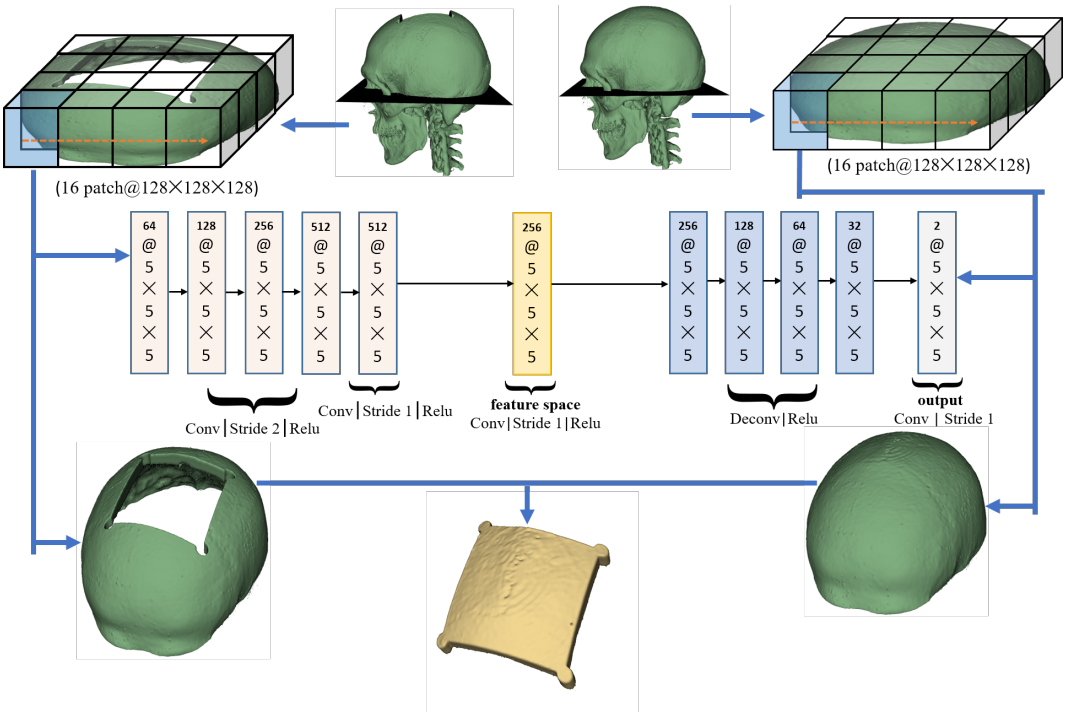

Figure 4.1: The architecture of the patch-based auto-encoder network for the skull defect restoration and cranial implant generation.

contains training pairs from $\left\{S_{d1}, S_{g1}\right\}$ to $\left\{S_{dN_{tr}}, S_{gN_{tr}}\right\}$. S_{di} is the i^{th} defected skull, and S_{gi} is the corresponding ground truth (the complete skull). N_{tr} is the number of training pairs in the training set ($N_{tr} = 85$). Define $p_{dij} \in 128^3$, $p_{gij} \in 128^3 (i = 1, 2...N_{tr};\ j = 1, 2...N_p)$, where N_p is the number of patches, as the j_{th} patch extracted from s_{di} and s_{gi}. **EPOCH** is the number of training epochs, and we make sure that all the models can fully converge given the training epochs. The training strategies are summarized in Algorithm 1 and Algorithm 2 and illustrated in Figure 4.3.

Non-overlapping Cropping As the name suggests, we extract $N_p = 4 \times 4 = 16$ patches of dimension 128^3 from the skull ($512 \times 512 \times 128$), such that the patches do not overlap each other (see Figure 4.3 (b)). For $N_p = 16$ successive training epochs, the extracted patches are sequentially fed into

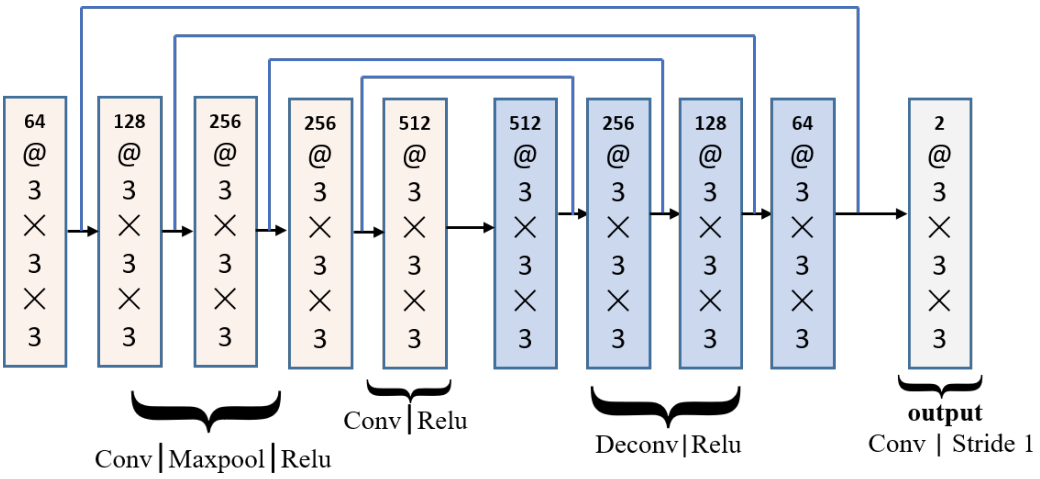

Figure 4.2: Added skip connections to the encoder-decoder network.

Algorithm 1 (Non)overlapping Cropping

1: model initialization θ^0
2: **for** $epoch \leftarrow 1$ to **EPOCH do**
3: $\quad i = random(1, N_{tr})$
4: $\quad$ **for** $j \leftarrow 1$ to $\mathbf{N_p}$ **do**
5: $\quad\quad \theta^{\mathbf{epoch},j+1} = model\_update(\theta^{\mathbf{epoch},j}, p_{dij}, p_{gij})$
6: $\quad$ **end for**
7: **end for**
8: $\theta^* = \theta^{\mathbf{EPOCH}}$

Algorithm 2 Random Cropping

model initialization θ^0
for ***epoch*** $\leftarrow 1$ to **EPOCH do**
  $i = random(1, N_{tr})$
  $x = random(1, 512 - 128)$
  $y = random(1, 512 - 128)$
  $p_{di} = s_{di}[x : x + 128, y : y + 128, :]$
  $p_{gi} = s_{gi}[x : x + 128, y : y + 128, :]$
  $\theta^{\mathbf{epoch}+1} = model\_update(\theta^{\mathbf{epoch}}, p_{di}, p_{gi})$
end for
$\theta^* = \theta^{\mathbf{EPOCH}}$

the network to update the network parameters, one patch for each epoch. After the $N_p = 16$ successive training epochs, another skull data is selected, and the same procedure is applied.

Overlapping Cropping The overlapping cropping training strategy is similar to the non-overlapping cropping. The difference is that for overlapping cropping, we extracted $N_p = 7 \times 7 = 49$ patches from each skull, and the patches are overlapped in the middle (see Figure 4.3 (a)). The network parameters are updated for $N_p = 7 \times 7 = 49$ successive training epochs by these patches, before another skull data is selected.

Random Cropping For the conventional random cropping, each training epoch utilizes a 128^3 patch randomly cropped from the skull and then switches to another skull data set.

For ease of reference, the networks in Figure 4.1 and Figure 4.2 are referred to as $Model_1$ and $Model_2$ respectively. We denote the model trained using non-overlapping cropping ($-n$), overlapping cropping ($-o$) and random cropping ($-r$) strategy as $Model - n$, $Model - o$ and $Model - r$. For example, $Model_1 - n$ denotes the $Model_1$ trained using the non-overlapping strategy. We will demonstrate that, the models trained using the proposed training strategies (Algorithm 1) perform significantly better than the random cropping, which is widely used in other studies.

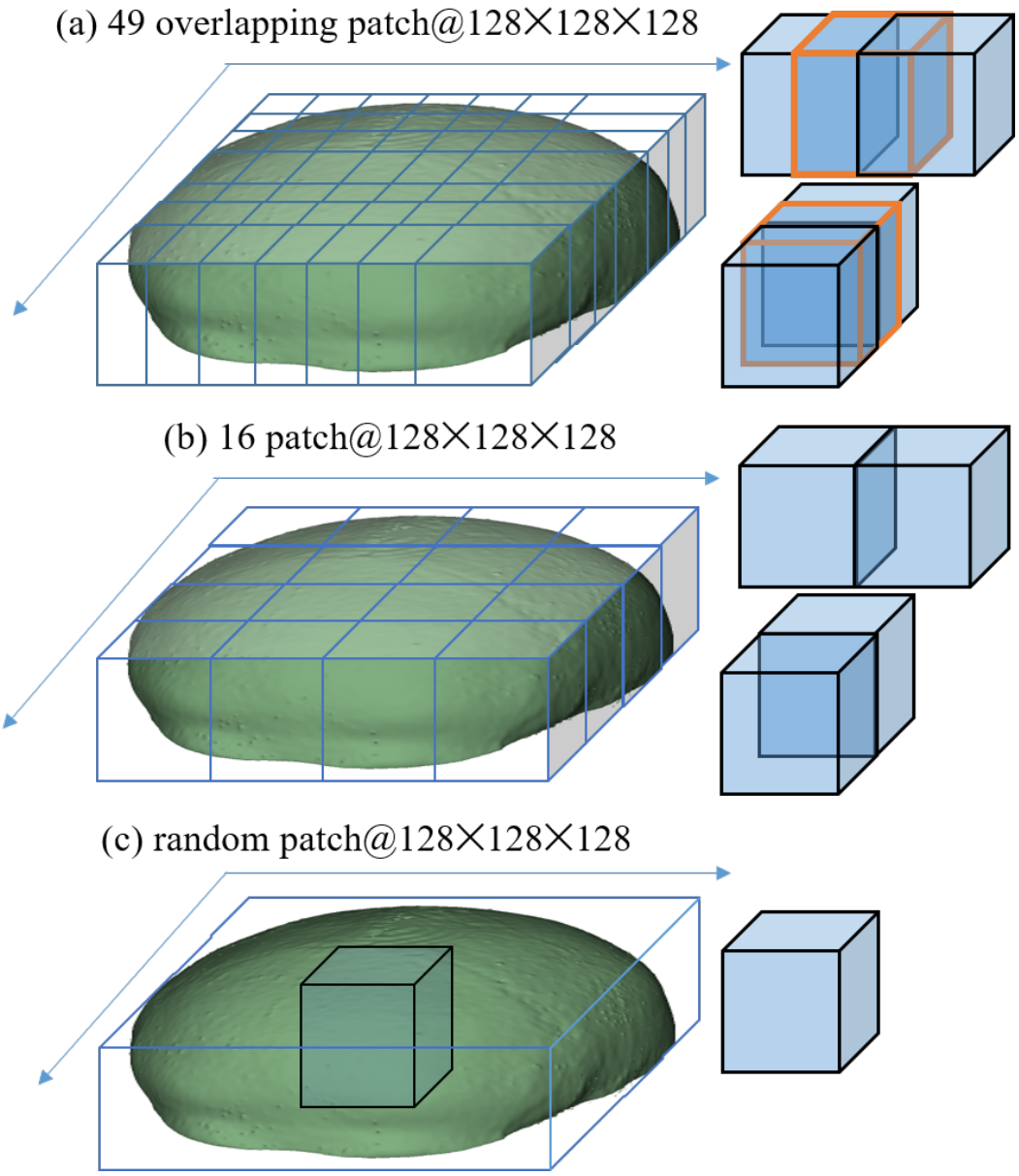

Figure 4.3: Illustration of overlapping (a), non-overlapping (b) and random cropping training strategy (c).

5 Experiment and Results

5.1 Experiments

5.1.1 Implementation

We implemented $Model_1$ and $Model_2$ using Tensorflow, on a machine with an Intel(R) Core(TM) i5-6600K CPU and a Nvidia GeForce GTX 1070Ti GPU (8GB GDDR5). The loss function is defined in Equation (4.4), and the optimizer is Adam. The training ($85 \times 9 = 765$) and testing ($82 \times 9 = 738$) sets are described in **Chapter 3** (B). For the experiments, we train $Model_1$ from scratch (weights initialized from normal distribution), using non-overlapping cropping, overlapping cropping and random cropping, respectively, and obtain the trained models $Model_1 - n$, $Model_1 - o$ and $Model_1 - r$. $Model_2$ is also trained from scratch (initial weights drawn from normal distribution) using non-overlapping cropping, and the trained model is $Model_2 - n$. Each training takes approximately one week.

5.1.2 Evaluation Metrics

We evaluated the trained models in terms of Dice Similarity Coefficient (DSC): $DSC = 2|G \cap P| / (|G| + |P|)$ (P: the prediction, G: the ground truth) and Jaccard Similarity Coefficient (JSC, also known as Intersection Over Union (IOU)): $|P \cap G| / (|P| + |G| - |P \cap G|)$. These metrics are commonly used to measure the similarity between two binary volumetric masks. 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): $PPV = TP/(TP + FP)$ and recall (also known as sensitivity, true positive rate): $Recall = TP/(TP + FN)$. From this perspective, the models are considered as binary classifiers. The AUC (area under the ROC curve) is also adopted for a more comprehensive evaluation of the classification ability of the models. We calculate these metrics for both the skull (the reconstructed skull and the skull ground truth) and the implant (the implant obtained by subtracting the defected skull from the reconstructed skull, and the ground truth implant). For convenience, DSC and JSC are referred to as **reconstruction accuracy**, and Precision and Recall are referred to as **classification accuracy**. For a more advanced evaluation, we created the surface model (triangular mesh) for some selected test cases and calculated the Hausdorff Distance (HD) w.r.t. the mesh created from the ground truth G_m and the mesh created from the prediction P_m:

$$HD(G_m, P_m) = max\left\{d(G_m, P_m), d(P_m, G_m)\right\}, \tag{5.1}$$

where $d(G_m, P_m) = max\left\{min\left\|p_G - p_P\right\|_2\right\}$; $p_G \in G_m$ are the vertices belonging to G_m, and $p_P \in P_m$ are the vertices belonging to P_m. For a qualitative evaluation and comparison, we also show the reconstructed skull and the implant in both 2D and 3D. The HD between the reconstructed triangular mesh and the ground truth is color-coded and visualized as a red-green-blue colormap (red indicates small errors, and blue indicates large errors).

5.2 Results

We first performed skull reconstruction on the $82 \times 9 = 738$ test cases using the trained models $Model_1 - n$, $Model_1 - o$, $Model_1 - r$ and $Model_2 - n$. For each test case, we obtained the implant by subtracting the defected skull from the reconstructed skull according to Equation (4.2). The implant and the reconstructed skull are compared against the ground truth to calculate the evaluation metrics, and the results are shown in Table 5.1.

We observe from Table 5.1 that $Model_1 - n$ and $Model_1 - o$ outperform $Model_1 - r$ 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 ($-r$). $Model_2 - n$ has the best performance among the trained models, and all the metrics regarding the skull exceed 0.90 (DSC: 0.9508, Precision: 0.9622, Recall: 0.9403, JSC: 0.9075), which is a significant improvement compared to $Model_1$ for both reconstruction accuracy and classification accuracy. The performance of $Model_2$ shows the advantage of using skip connections in the encoder-decoder network. Note that $Model_2$ is only half the size of $Model_1$ in terms of the number of trainable parameters. Compared with the skull, the metrics for the implant are much lower. Some of the poor results may come from the failure of the models when the defect is very large and the defect is generated against the lower border of the cropped skull ($512 \times 512 \times 128$), where the context information is missing. For the majority of the training and testing cases, the defects are generated within the skull where the defect context is available (see Figure 3.2 (b)). The poor results regarding the metrics can be primarily attributed to the mismatch between the reconstructed skull and the ground truth skull, which is represented as noise in the implant obtained by subtraction based on Equation (4.2). We will further explain the issue in later sections. As we stated in **Chapter 3** (B), we considered nine different defects for each of the 82 unique skulls in the test set. For all the metrics, we take the mean of the nine scenarios as the final score for each unique skull. The scores of the 82 cases are given as boxplots in Figure 5.1 (the skull) and Figure 5.2 (the implant). We can see that $Model_2 - n$ has the best performance, while $Model_1 - r$ has the worst performance. Besides, it can be seen that the performance of $Model_2 - n$ is more stable among the test cases compared to other models, as the scores for $Model_2 - n$ are more concentrated in the boxplot.

Figure 5.3 shows a *skull/implant* scatter plot, and we can see that the skull and the implant are positively related in terms of reconstruction accuracy (DSC and JSC) and classification accuracy (Precision and Recall), which is in accordance with the demonstration in **Chapter 4** (A): Minimizing the skull error is equivalent to minimizing the implant error.

The ROC curve in Figure 5.4 shows the discriminative ability of the four trained models when considered as binary classifiers. This curve is created based on the probabilistic output of the models and the ground truth skulls for the nine cases in Figure 5.5 at various threshold settings. The probabilistic

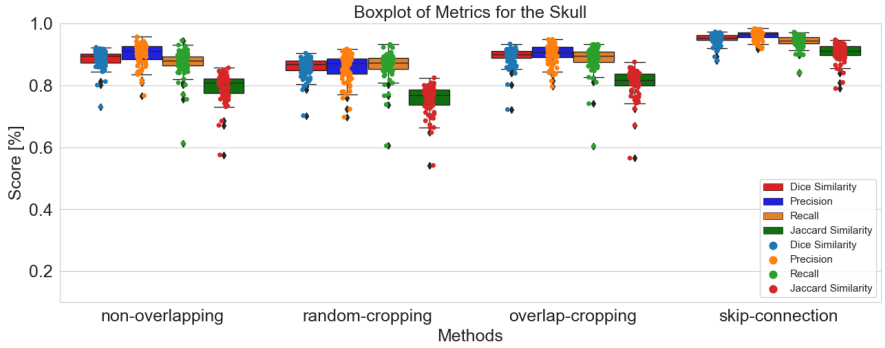

Figure 5.1: Boxplot of the metrics for the skulls by non-overlapping cropping ($Model_1 - n$), random cropping ($Model_1 - r$), overlapping cropping ($Model_1 - o$) and skip connection ($Model_2 - n$).

output of the models gives the probability of an voxel belonging to the unoccupied space (the background and the empty space inside the skull) and the occupied space (the skull). We can see from the AUC (the area under the ROC curve) value that all the four models are strong classifiers w.r.t. the nine cases in Figure 5.5, whereas $Model_1 - o$ and $Model_2 - n$ performs better than $Model_1 - n$ and $Model_1 - r$.

5.2.1 Statistical Significance Analysis

Via t-tests, we further analysed whether the improvement of using skip connections ($Model_2 - n$) and the proposed training strategy ($-n$, $-o$) is statistically significant regarding all the metrics for both the skull and the implant. We calculated $t = (\bar{X}_1 - \bar{X}_2)/s_p\sqrt{\frac{2}{n}}$, where $s_p = \sqrt{\frac{s^2_{X_1} + s^2_{X_2}}{2}}$ and $\bar{X}_1$, $\bar{X}_2$,$s^2_{X_1}$,$s^2_{X_2}$ are the mean and the estimated variance w.r.t. a metric from two trained models. n is the number of test cases. Table 5.2 shows the results. In this table, *Model* is abbreviated as M.

We adopt the commonly used $P_T = 0.5$ as the threshold to categorize the difference as significant or insignificant. Table 5.2 shows that $Model_2 - n$ performs significantly better than $Model_1 - o$, $Model_1 - n$ and $Model_1 - r$ for both the skull and the implant regarding most of the metrics. The

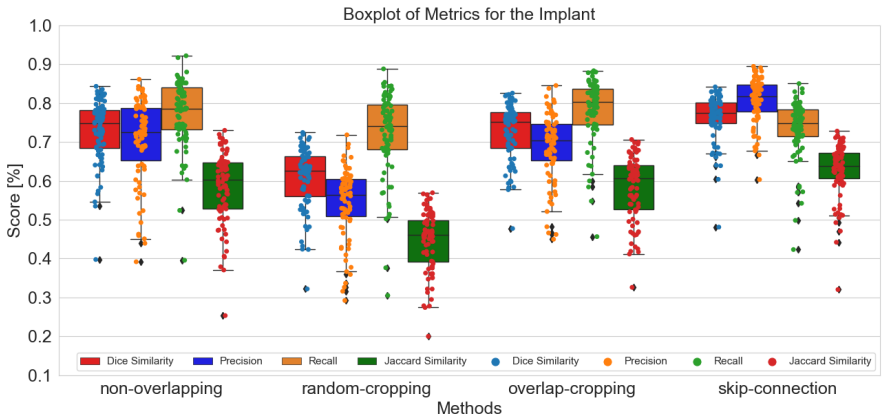

Figure 5.2: Boxplot of the metrics for the implants by non-overlapping cropping ($Model_1 - n$), random cropping ($Model_1 - r$), overlapping cropping ($Model_1 - o$) and skip connection ($Model_2 - n$).

proposed training strategy ($Model_1 - n$ and $Model_1 - o$) also sees significant improvements compared to a conventional training strategy ($Model_1 - r$). We see no significant difference between $Model_1 - n$ and $Model_1 - o$ in terms of the metrics except the recall ($Model_1 - o$ performs slightly better than $Model_1 - n$ regarding the metrics according to Table 5.1). Nonetheless, their difference can be better observed through visual inspection of the reconstruction in Figure 5.5.

5.2.2 3D Visualization

For a visual inspection, we show the hole-filling results for nine different skulls with defects of different shapes, positions and sizes in Figure 5.5. The reconstruction results in the third to fifth row are produced by $Model_1 - o$, $Model_1 - n$ and $Model_1 - r$, respectively. The 3D view of the reconstruction results shows that the defects can be restored by the three approaches, but $Model_1 - n$ and $Model_1 - o$ generally perform better than $Model_1 - r$ w.r.t. the reconstruction quality. The top right of each image shows the 2D sagittal view of the defected skull (red) overlapped with the reconstructed skull (white). On the one hand, from the 2D views, we can see how the borders

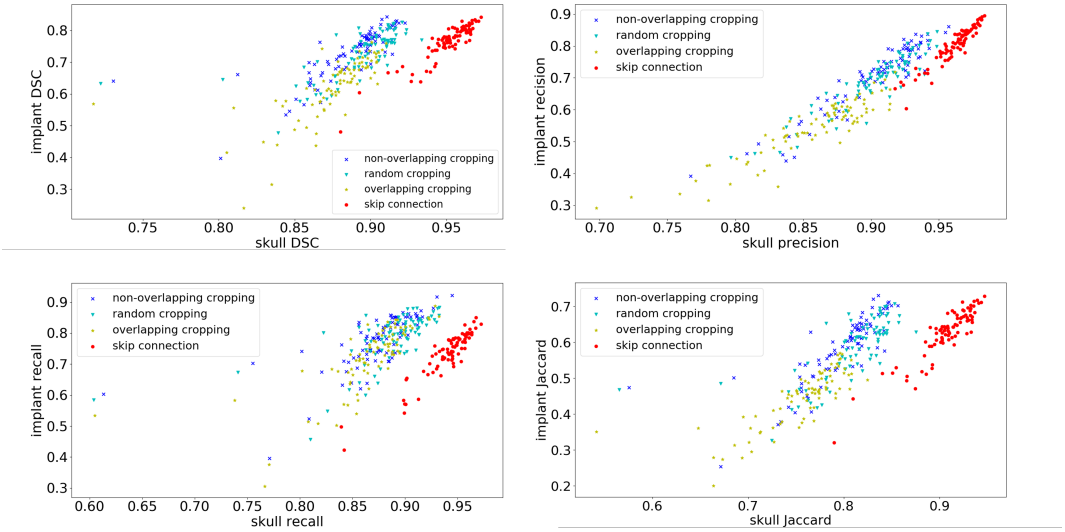

Figure 5.3: Scatter plot of *skull*/*implant* regarding the metric by the four trained models.

match between the defected area and the restored defect (i.e., the implant). On the other hand, the hole-filling model is expected to restore the defect, while maintaining as much as possible the original structure of the defected skull. This can also be seen from the 2D views outside the defected area on the skull (only the upper part of the skull ($512 \times 512 \times 128$) needs to be inspected). We can see that the reconstructed skull (white) in the 2D view apart from the defected area indicates a good overlap (match) with the defected skull (red).

For cranial implant design in cranioplasty, the borders between the defect and the implant are most critical concerning the protection the implant can provide to the brain inside. The first to third row in Figure 5.6 shows the Hausdorff Distance (HD), represented by a red-green-blue colormap, between the triangular mesh of the ground truth and the reconstructed skull produced by $Model_1 - o$, $Model_1 - n$ and $Model_1 - r$. The HD is calculated w.r.t. the meshes created from the ground truth and from the reconstruction according to Equation (5.1). Column one to nine correspond to the nine cases in Figure 5.5. In the red-green-blue colormap, red indicates a small error while blue indicates a large error. For each case (column), the colormap is adjusted to the same color range, so that we can see the level of mismatch

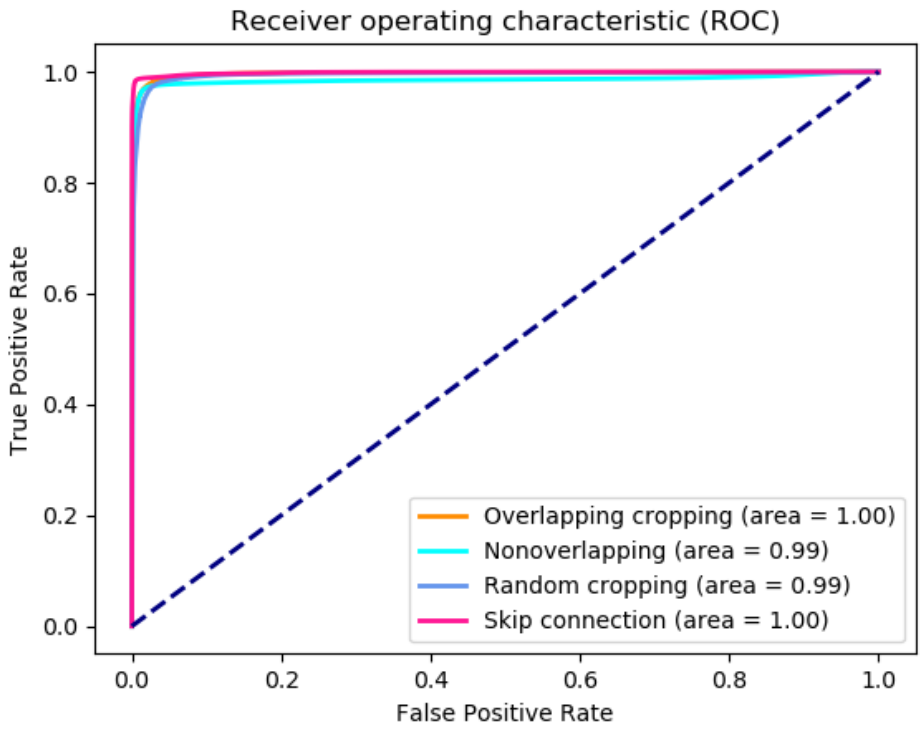

Figure 5.4: Receiver operating characteristic (ROC) curve of the four models created based on the nine cases in Figure 5.5.

of a certain area on the skull simply by looking at the color. On the left side of each colormap, the histogram of the HD values is shown, and the concentration of the HD values near zero indicates an overall small error. It can be seen that $Model_1 - r$ has the worst performance, which is in accordance with the observation from Figure 5.5. We can also see from the colormap that there are different level of mismatch between the ground truth and the prediction throughout the skull. Generally, the largest mismatch between the ground truth and the reconstructed skull comes from the restored region (the implant). The fourth row in Figure 5.6 shows the HD colormap between the mesh created from the ground truth implant and the mesh created from the implant obtained by subtracting the defected skull from the reconstructed skull (by $Model_1 - n$). The border and the fine

details (the roundish corner) of the implant in the colormap are mostly red, indicating a small distance (error) and that the implant can fit precisely against the border of the defected region. The last row shows the implant (without post-processing). We can see that the mismatch between the ground truth and the reconstruction throughout the skull can lead to a *noisy* implant, and the largest distance (error) comes from the *noise* which can not be removed automatically using connected component analysis. The *noise* is also the primary cause of the poor results for the implant in Table 5.1.

Figure 5.7 shows the skull reconstruction (first row), HD color map of the skull (second row) and the implant (third row) by $Model_2 - n$ for the same nine cases in Figure 5.5 and Figure 5.6. We can observe that the behavior of $Model_2 - n$ is distinct from $Model_1 - o$, $Model_1 - n$ and $Model_1 - r$: $Model_2 - n$ achieves a much greater performance of reconstructing the original structure of the defected skull, so that the mismatch between the reconstructed skull and the ground truth is close to zero. This can be confirmed from the HD colormap in Figure 5.7. Therefore, we can obtain a *clean* implant via subtraction (the third row) compared with the *noisy* implant produced by the other three models. Nonetheless, the performance of $Model_2 - n$ in restoring the defected area is not as optimal as in reconstructing the original skull structure, as can be seen from Figure 5.7. However, the fine detail of the implant (e.g., the small roundish corner) can still be reconstructed.

In Table 5.3, we summarize the statistics, i.e., the max, mean and root mean square (RMS), of the HD values of skull reconstruction by $Model_1 - o$, $Model_1 - n$, $Model_1 - r$ and $Model_2 - n$ for the nine cases in Figure 5.6 and Figure 5.7.

5.2.3 Post-processing for 3D printing

After reconstruction of the skull by the models, the restored defect (i.e., implant) can be obtained by subtracting the defected skull from the reconstructed skull according to equation (4.2). As we discussed previously, the *noisy* implant needs to be converted into STL format which can easily be done using ITK-SNAP or 3D Slicer, and the *noise* needs to be removed

manually using MeshLab. Alternatively, the *noise* can be removed first (e.g., using 3D Slicer) before converting it into STL format. The manipulation is easy and usually takes one to two minutes, after which the *cleaned* STL file is ready for 3D printing. Figure 5.8 shows a comparison between the implant and the HD colormap before and after noise removal, which can lead to an increase of the evaluation metrics (e.g., the DSC) and a decrease of the mean HD error.

Figure 5.9 shows an example of implant cleaning, STL conversion and a 3D printed implant.

5.2.4 Learnt Hidden Representations and Robustness Analysis

To have a better interpretation of how the model performs the hole filling (defect restoration) task and understand the mechanism behind the difference among different training strategies in their performance, we visualized the activation maps of the first and second convolutional layers of the trained model $Model_1 - o$ (Figure 5.10 (b)), $Model_1 - n$ (Figure 5.10 (c)) and $Model_1 - r$ (Figure 5.10 (d)). Higher order activation maps are generally abstract and incomprehensible to humans, so only the first two layers are considered. The input is a 128^3 patch cropped from the defected skull shown in Figure 5.10 (a). We considered two types of defects: the left defect with small roundish corner and the right defect of the same skull without the roundish corner. Note that the type of defect on the right is not involved in training the models. For both types of defect, we have experimented with the middle four patches (marked as (1),(2),(3) and (4)), which contains the defected area to create the activation maps. Here we only use the first patch (marked (1)) for illustration, as we observed similar activation patterns for the rest of the patches. For ease of reference, we denote the selected patch on the left and right image of Figure 5.10 (a) as p_{d1} and p_{d2}. We define the activation maps of the first and second convolutional layer as O^1_{64} (dimension: 64^3) and O^2_{128} (dimension: 32^3). 64, 128 is the number of activation maps. Figure 5.10 (b) (c) and (d) show some of the feature maps in O^1_{64} and O^2_{128}, produced by $Model_1 - o$, $Model_1 - n$ and $Model_1 - r$. The first four images in each figure show the activation maps given as input p_{d1},

and the last four images show the activation map given as input p_{d2}. We can see from Figure 5.10 (b) (c) (d) that the first activation map highlights the edges of p_{d1}. The second activation map (to the right) shows that the unoccupied voxels in p_{d1}, which are in the background and the defected region, are detected and segmented (the high activations in yellow). The third map shows the discrimination of the background and the target, where the unoccupied voxels should be filled, by highlighting the edges (in yellow) of only the defected region. From the visualization, we can see that some of the convolutional filters in the first two layers function as robust edge detectors. The last four activation maps in Figure 5.10 (b) (c) (d) are from p_{d2} and follow a similar pattern as the first four activation maps from p_{d1}.

Based on these observation, we may assume how the trained models perform the hole filling task: ***1)*** detect edges – ***2)*** detect unfilled regions (the background and the defected region) – ***3)*** distinguish between the unfilled regions and identify the target (the target to be filled is only the defected region) – ***4)*** fill the target. The last step, ***fill the target***, requires that the thickness of the restored defect be consistent with the defected area at the border, and this sophisticated process must be performed by higher-order convolutional operations. Note that, in the first three steps, the fine detail of the defect (the small roundish drilling holes in the corner) is captured. The purpose of the experiment is threefold. *First*, understand the hole-filling mechanism of the deep learning models. *Second*, compare the three approaches from the perspective of the learned features and understand the difference of their hole-filling performance. As can be seen from Figure 5.10 (b) (c) and (d), all three models capture useful features of the input patch, such as the edges and the the region of unoccupied voxels. However, their ability to distinguish between the background and the defected area differs. From the second and fourth row of Figure 5.10 (b) (c) and (d), we can see that the edges of the defected area captured by $Model_1 - r$ are the weakest compared with $Model_1 - o$ and $Model_1 - n$, which subsequently leads to the worst hole-filling performance among the models. *Third*, evaluate the robustness and generalization performance of the models w.r.t. defects of unknown shapes such as p_{d2}. We show in the third and fourth row of Figure 5.10 (b) (c) and (d) the activation maps given as input p_{d2}. We see that the edges and the unoccupied regions are still captured despite the change to the input. We expect a robust model to be able to restore any defect,

regardless of the defect shape, position and size. From the perspective of feature embedding, defects of different shapes should be encoded into the feature space such that their representations are highly correlated, if a model is robust. To illustrate the concept, we calculate the Pearson correlation coefficient between the encoded representations of p_{d1} and p_{d2}, produced by the feature layer of $Model_1 - n$ (the sixth layer in Figure 4.1). The encoded representation is denoted as $X = O^6_{256} \leftarrow p_{d1}$, and $Y = O^6_{256} \leftarrow p_{d2}$. 256 is the number of feature maps of dimension 8^3. The correlation coefficient between X and Y can be calculated according to equation (5.2):

$$\rho(X_i, Y_j) = \frac{cov(X_i, Y_j)}{\sigma_{X_i}\sigma_{Y_j}}, i, j = 1, 2, ..., 256 \tag{5.2}$$

cov denotes the covariance, and σ denotes the standard deviation.

The mean correlation coefficient of the 256 feature maps is 0.9796, indicating that the encoded representations of two different defects (p_{d1} and p_{d2}) are highly correlated, and, thus, the model is robust and can generalize across different defect shapes.

For a better illustration of the high correlation, we calculate the encoded representation of p_{d1} and p_{d2} for $Model_1 - o$, $Model_1 - n$ and $Model_1 - r$. For $Model_2 - n$, we choose the output of the fifth layer (see Figure 4.2) as the encoded representation, which has 512 feature maps of dimension 8^3. For visualization, the representations are further embedded into a two-dimensional space using uniform manifold approximation and projection (umap) (McInnes and Healy, 2018). The embedded scatter points are shown in Figure 5.11, and the blue and red points come form p_{d1} and p_{d2}, respectively. It is easy to see that the embedded feature maps from p_{d1} and p_{d2} follow a very similar distribution in the two-dimensional space.

Figure 5.12 shows the probability map produced by the last layer of $Model_1 - n$ (Figure 5.12 (a),(b)) and $Model_2 - n$ (Figure 5.12 (c),(d)) given as input the defected skull in Figure 5.10 (a). The probability maps are of dimension $512 \times 512 \times 128$ (same as the defect skull) and shows the confidence of the model in assigning a voxel as being occupied in the skull and the implant. The last image of each image shows a sagittal slice of the probability map of the skull. We can see that $Model_1 - n$ predicts all occupied voxels with very

high confidence (over 0.96), except the skull surface (green). $Model_1 - n$ predicts the implant for the defected region with high confidence, which indicates a high tolerance of the model to the input disturbance (e.g., the change of defect shape, noise, etc). In contrast, the prediction confidence for $Model_2 - n$ is much lower (0.67), indicating that $Model_2 - n$ might be more sensitive to the input disturbance.

5.2.5 Clinical Applicability

To evaluate the applicability of the methods in a real clinical scenario, we collected the CT scans of the head from craniotomy. The skull was manually opened by neurosurgeons during craniotomy, leaving a defect in the skull. For protective purpose, the defect was temporarily filled by a mesh-like thin layer made of titanium. Figure 5.13 (a) shows such a CT scan where the thin titanium layer is visible. The skull was then segmented from the CT scan and Figure 5.13 (b) shows the segmented skull in 3D, axial, sagittal and coronal view. It is obvious that the defect from craniotomy is much more complex than the synthetic defects used in training the models, and the existence of the temporary protective layer on the defected area can also potentially affect the model performance. However, the models are still able to perform defect restoration in such case despite the complexity. Figure 5.13 (c-e) shows the results produced by $Model_1 - o$, $Model_1 - n$ and $Model_1 - r$. We can see from the 3D view that all the models successfully complete the skull volume.

We also report a case of failure in Figure 5.14, where the defect from craniotomy is too large and complex for the model to restore. From Figure 5.14 (b), we can see that the large defect is partially covered by a protective layer and occupies almost one-third of the entire skull. For this case, the defect is only partially filled by the models as is shown in Figure 5.14 (c-e). Note that both of the two cases from craniotomy are distinct from the data in model training.

A ground-truth implant or pre-operative CT scan of the head for the two cases were not available. For a quantitative comparison, the defected skull and the reconstructed skulls are converted into 3D triangular meshes, and

we calculate the HD values between the two meshes as in Figure 5.6. Figure 5.15 (b) and (d) show the HD colormap between the defected skull mesh and the reconstructed skull mesh from $Model_2 - o$ (left), $Model_2 - n$ (middle) and $Model_2 - r$ (right).

5.2.6 Reproducibility

The datasets and the code involved in this study are scheduled to be made public shortly. We provide an interactive environment for the skull defect restoration functionality in Studierfenster (`http://studierfenster.tugraz.at/`), where users can interact with the algorithm using the sample data we provide. A Youtube video is available at `https://www.youtube.com/watch?v=pt-jw8nXzgs` for a quick preview of the concept of automatic skull defect restoration and cranial implant generation.

Table 5.1: Mean values of the Dice Similarity Score (DSC), the Jaccard Similarity Coefficient (JSC), Precision and Recall of the $82 \times 9 = 738$ testing cases for the skulls and implants.

	skull				**implant**			
	DSC	JSC	Precision	Recall	DSC	JSC	Precision	Recall
$Model_1 - n$	0.8861	0.7971	0.9023	0.8726	0.7276	0.5830	0.7042	0.7739
$Model_1 - o$	0.8931	0.8086	0.9020	0.8863	0.7267	0.5810	0.6888	**0.7838**
$Model_1 - r$	0.8598	0.7559	0.8573	0.8655	0.6080	0.4457	0.5437	0.7210
$Model_2 - n$	**0.9508**	**0.9075**	**0.9622**	**0.9403**	**0.7633**	**0.6257**	**0.8059**	0.7346

Table 5.2: P values ($P_T = 0.5$) among the evaluation metrics of the four approaches for statistical significance analysis.

	skull				**implant**			
	DSC	Precision	Recall	JSC	DSC	Precision	Recall	JSC
$M_2 - n \leftrightarrow M_1 - o$	**1.5184e-34**	**3.7298e-35**	**2.2795e-18**	**1.6861e-38**	**0.0005**	**1.0105e-18**	**8.4863e-05**	**0.0003**
$M_2 - n \leftrightarrow M_1 - n$	**4.5111e-40**	**7.4560e-30**	**5.4807e-26**	**4.3297e-44**	**0.0014**	**2.2395e-12**	**0.0023**	**0.0011**
$M_2 - n \leftrightarrow M_1 - r$	**4.7962e-55**	**6.8016e-48**	**1.0148e-28**	**2.8876e-60**	**4.7696e-30**	**2.1124e-50**	0.3464	**4.8481e-34**
$M_1 - n \leftrightarrow M_1 - r$	**6.6451e-08**	**1.1635e-11**	0.2902	**1.2935e-08**	**2.3080e-17**	**9.2653e-20**	**0.0006**	**2.3803e-19**
$M_1 - o \leftrightarrow M_1 - r$	**2.9863e-11**	**1.5957e-12**	**0.0027**	**2.1268e-12**	**1.8597e-18**	**1.0788e-19**	**3.6496e-05**	**1.6586e-20**
$M_1 - o \leftrightarrow M_1 - n$	0.1221	0.9448	**0.0438**	0.0972	0.9411	0.3116	0.4484	0.8839

Figure 5.5: 3D visualization of skull reconstruction by $Model_1 - o$ (third row), $Model_1 - n$ (fourth row) and $Model_1 - r$ (fifth row). The first row shows the defected skull to be reconstructed (the input), and the second row shows the ground truth. The 2D sagittal views on the right corner of each image show how the border between the defected skull (red) and the reconstructed skull (white) match.

Figure 5.6: HD in red-green-blue colormap between the prediction (first row: $Model_1 - o$, second row: $Model_1 - n$ and third row: $Model_1 - r$) and the ground truth for the nine cases in Figure 5.5. Fourth row: the HD colormap between the implant (by $Model_1 - n$) and the ground truth. The left side of each colormap shows the histogram of the HD values. Fifth row: the implant (by $Model_1 - n$) from the subtraction of the defected skull from the reconstructed skull.

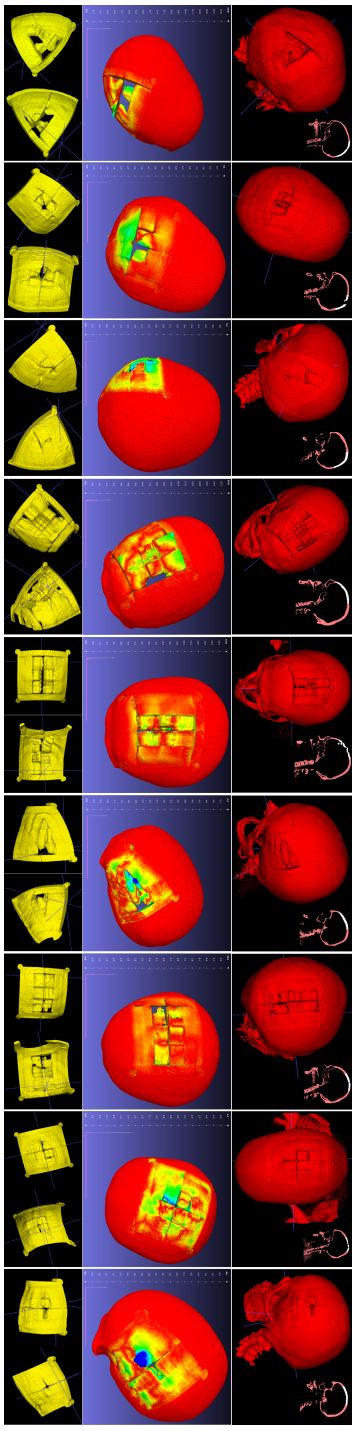

Figure 5.7: The 3D reconstruction (first row), HD colormap (second row) and the implant (thrid row) for the nine cases in Figure 5.5 by $Model_2 - n$. The HD colomap has the same color range as that in Figure 5.6.

Table 5.3: Statistics of the Hausdorff Distance for the nine cases in Figure 5.6 and Figure 5.7. $-o$, $-n$, $-r$, $-s$ denotes overlapping cropping ($Model_1 - o$), non-overlapping cropping ($Model_1 - n$), random cropping ($Model_1 - r$) and skip connection ($Model_2 - n$). The minimum HD value is always zero.

	1	2.	3.	4	5	6	7	8	9
$Max - o$	**5.02**	11.11	4.32	**6.26**	7.00	**3.64**	**6.54**	**3.26**	5.06
$Max - n$	8.18	**4.87**	**3.68**	7.05	**4.35**	3.73	7.87	3.88	4.51
$Max - r$	11.99	48.17	4.10	12.31	11.61	3.93	13.01	30.30	**4.07**
$Max - s$	14.79	9.12	5.14	15.99	15.03	5.39	10.14	4.68	8.65
$Mean - o$	0.35	0.68	0.33	0.50	**0.32**	0.34	0.36	0.36	0.33
$Mean - n$	0.55	0.37	0.33	0.43	0.36	0.33	0.50	0.37	0.38
$Mean - r$	0.60	1.62	0.47	0.81	0.56	0.42	0.68	1.13	0.47
$Mean - s$	**0.30**	**0.27**	**0.12**	**0.36**	0.34	**0.10**	**0.28**	**0.19**	**0.12**
$RMS - o$	**0.55**	1.39	0.49	0.83	**0.56**	0.49	**0.57**	0.54	0.50
$RMS - n$	1.14	**0.52**	0.50	**0.72**	0.58	0.48	0.87	**0.51**	0.57
$RMS - r$	0.94	5.18	0.64	1.33	0.94	0.58	1.10	3.12	0.63
$RMS - s$	1.11	0.91	**0.48**	1.18	1.18	**0.38**	0.84	**0.51**	**0.48**

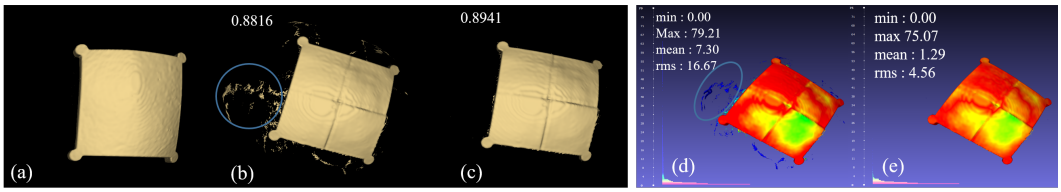

Figure 5.8: Manual cleaning of the implant. (a) The ground truth. (b) The prediction. (c) Manual removal of the noise in (b). The DSC improved from 0.8816 to 0.8941 after cleaning. (d) and (e) the Hausdorff Distance colormap before and after cleaning.

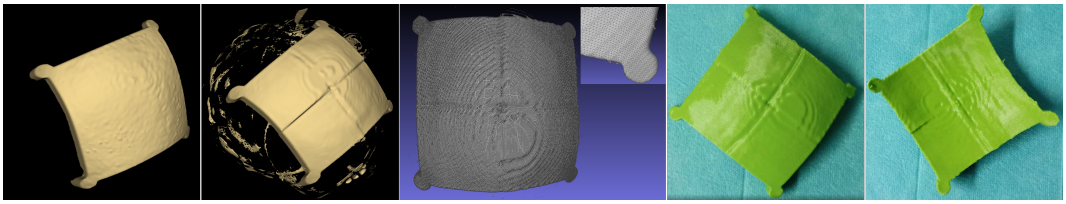

Figure 5.9: An example of a 3D printed implant (the last two images). The 3D printing material is the commonly used polylactic acid, which is biodegradable. From left to right: the ground truth implant, the implant obtained by subtracting the defected skull from the reconstructed skull, converting the implant data into the STL format and mesh cleaning, the 3D printed implant.

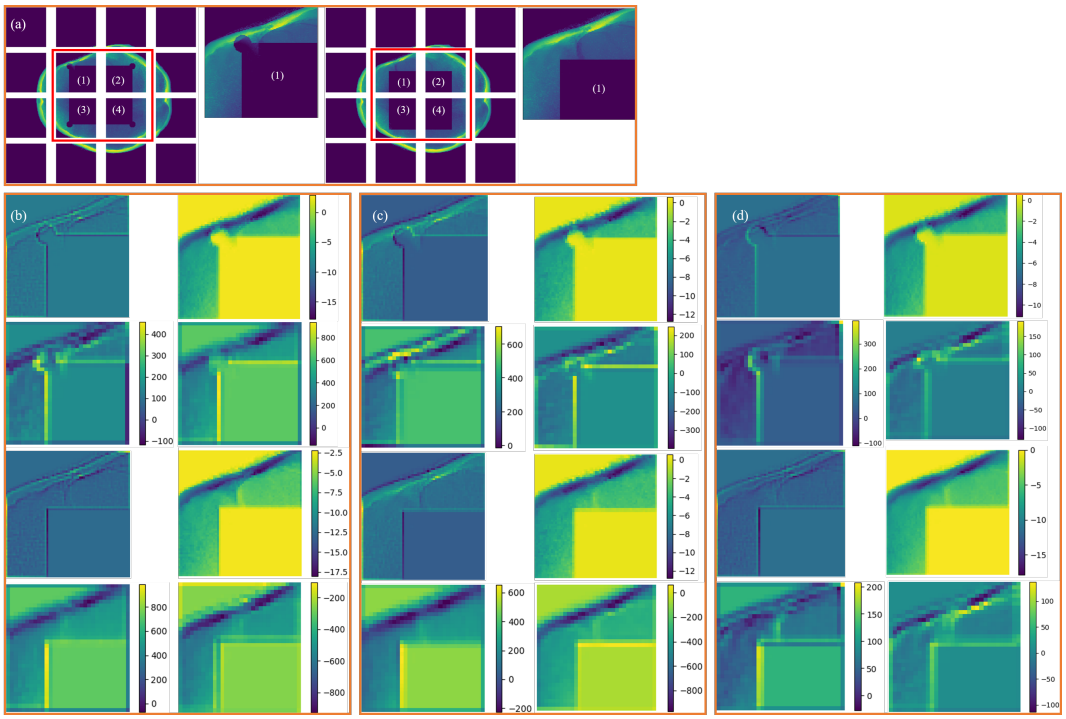

Figure 5.10: The activation maps of the first two convolutional layers produced by $Model_1 - o$ (b), $Model_1 - n$ (c) and $Model_1 - r$ (d), given as input p_{d1} and p_{d2}.

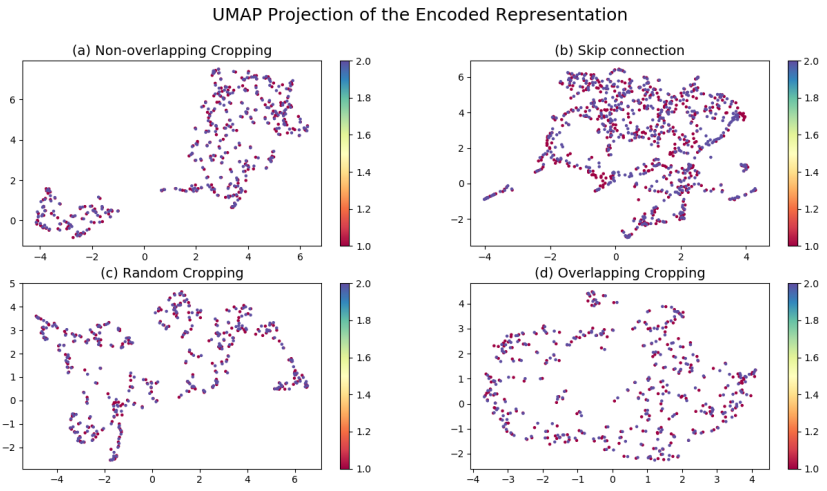

Figure 5.11: Mapping the encoded representation of p_{d1} and p_{d2} into two-dimensional space using Uniform Manifold Approximation and Projection (UMAP) in Euclidean distance.

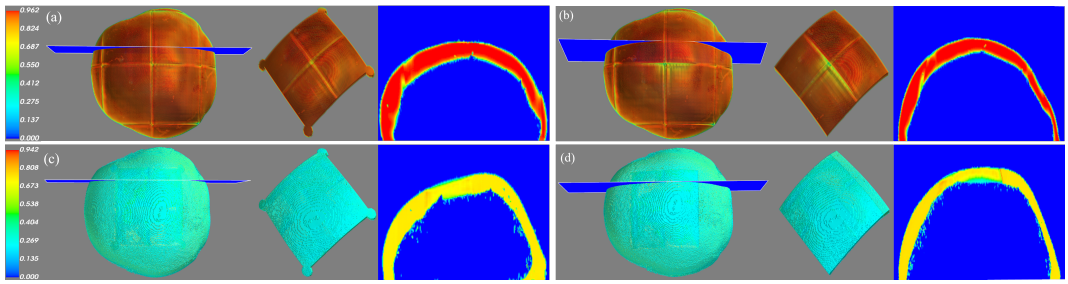

Figure 5.12: The probability map of the skull and the implant produced by the last convolutional layer, when given the skull in Figure 5.10 (a) as input. The last image in each figure shows a sagittal slice of the probability map, which visualizes the probability distribution inside the volume. (a), (b) are from $Model_1 - n$ and (c), (d) are from $Model_2 - n$.

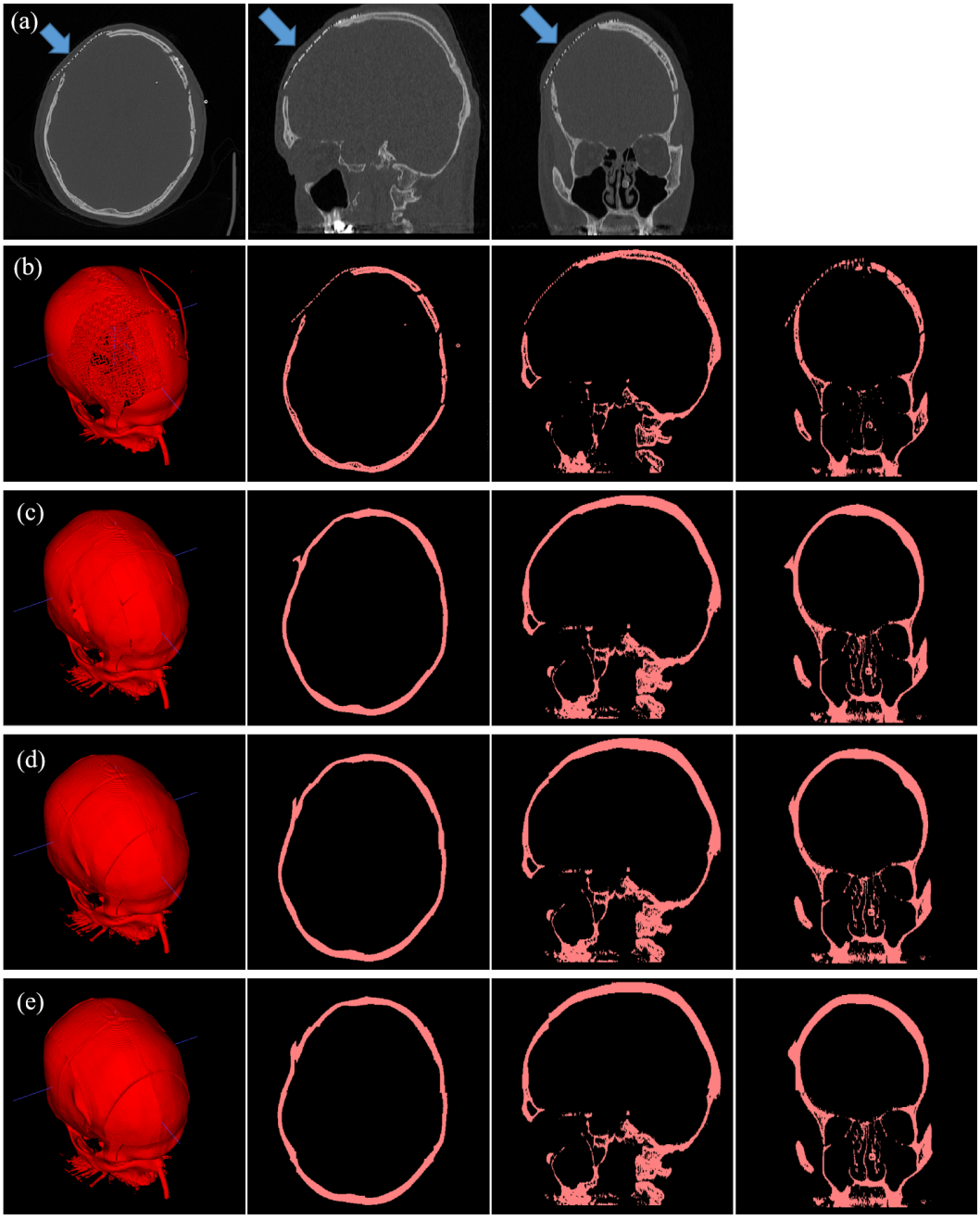

Figure 5.13: Reconstruction results from craniotomy data. (a) CT scan of a patient head from craniotomy. (b) Skull segmentation from the CT scan, viewed in 3D, axial, sagittal and coronal plane. Reconstruction results by $Model_1 - o$ (c), $Model_1 - n$ (d) and $Model_1 - r$ (e).

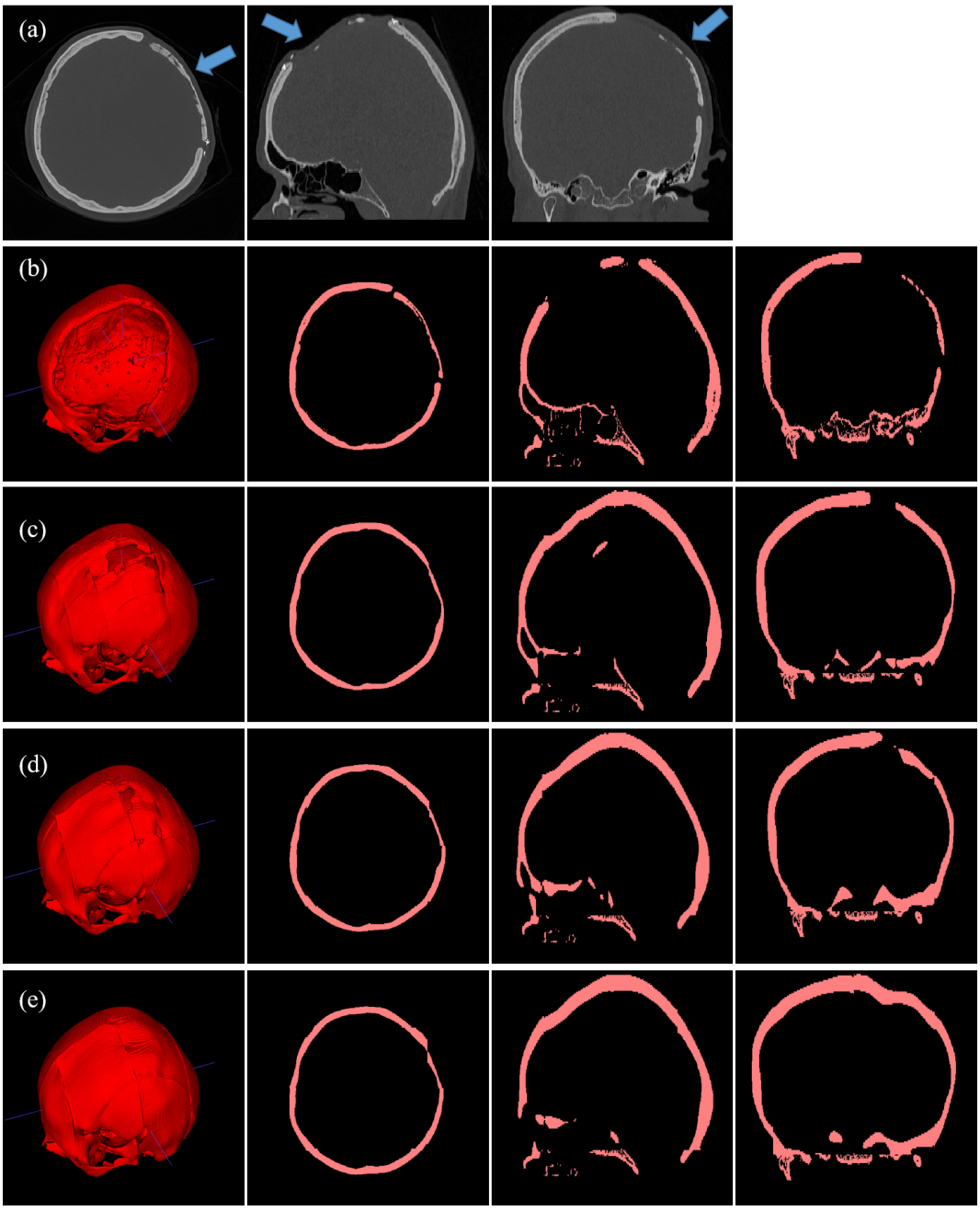

Figure 5.14: A case of failure from craniotomy with a very large defect. (a) the CT scan, (b) the skull segmentation from the CT scan and (c-e) the reconstruction results by $Model_1 - o$, $Model_1 - n$ and $Model_1 - r$.

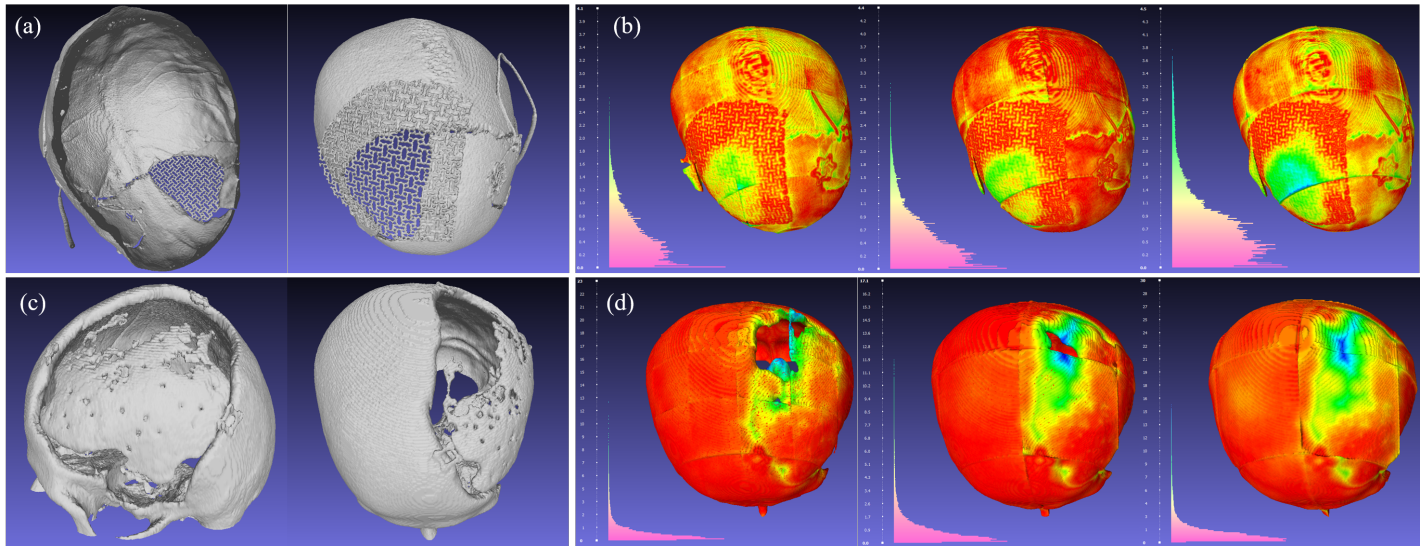

Figure 5.15: The Hausdorff distance between the defected skull mesh and the reconstructed skull mesh, represented by a red-green-blue colormap. (a) and (c) triangular mesh of the defected skull created from the segmented skull volume. (b) and (d) the Hausdorff distance w.r.t. vertex between the defected skull mesh and the reconstructed skull mesh by $Model_1 - o$ (left), $Model_1 - n$ (middle) and $Model_1 - r$ (right). The left side in figures (b) and (d) shows the histogram of the Hausdorff Distance.

6 Discussion

Patch-based training is a widely accepted strategy in deep learning when the data dimensionality is high. In a conventional patch-based scheme, the deep learning model trains on a sub-volume randomly cropped from the overall data set 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 structure of the target is not important. 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 dominant in tasks where the primary goal is to reconstruct the overall shape of the target, especially when the data is of sparse nature, like the skull. The skull data are spatially sparse and can be seen as a two-dimensional manifold embedded in three-dimensional volumetric space ($512 \times 512 \times Z$). The overall voxel occupancy rate of a skull in the volumetric space is usually no more than 10 percent. The spatial sparsity of the data can make it even harder for a deep learning model to learn useful features efficiently. The proposed training strategy overcomes the disadvantage of conventional patch-based training by extracting consecutive non-overlapping patches from the complete skull volume, which are then successively utilized for gradient computation in the optimization process. The whole skull volume can be involved in the optimization process before another skull volume is selected, so that the holistic shape distribution of the skull can be captured by the deep learning model. In the study, our computational resource restricts the patch dimension to 128^3. To deal with the sparsity issue, we proposed that using overlapping patches ($Model_1 - o$ and Figure 4.3 (a)) can reduce the *relative* sparsity of each single patch[1] without decreasing the actual patch size

[1]For non-overlapping cropping, the patch size is $(512 \times 512 \times 128)/(4 \times 4) = 128^3$. For overlapping cropping, the *relative* patch size is $(512 \times 512 \times 128)/(7 \times 7)$. The smaller the

and thus make it easier for the network to learn. The downside of using overlapping patch is that the computation will be increased (the larger the overlap, the less sparse the patch and the more computation needed).

patch, the less sparse relatively.

7 Conclusion and Future Work

We investigated the usability of deep learning for 3D volumetric shape completion, and the effectiveness is demonstrated in the field of automatic skull defect restoration and cranial implant design, which is highly desired in cranioplasty. We showed how a large skull database can be constructed and then utilized for training deep learning models. To work on high-resolution skull data, we proposed a tailored patch based strategy for training deep learning networks which shows a significant improvement compared to using a conventional training method ($Model_1 - o$ and $Model_1 - n$ perform better than $Model_1 - r$). The proposed training strategy is general and can be used in other applications involving training on high-resolution volume data, especially, when the data is of sparse nature, like the skull ($Model_1 - o$ performs slightly better than $Model_1 - n$). We also showed that using skip connections can lead to a remarkable increase in the performance of the deep learning model for the application ($Model_2 - n$ performs the best among the models). The deep learning models we built are interpretable by humans and shows robustness and good generalization performance w.r.t the shape of defect. We also tested the models using real defected skulls from craniotomy, and the results show promise for clinical applicability. Considering that a larger patch carries more overall shape information compared to a smaller patch, it is safe to assume that the larger the patch, the better the reconstruction performance of deep learning models, even if larger patches can bring about greater sparsity. Given the sparsity of the skull data, it can be expected that most of the computational and memory resources are wasted on the unoccupied area of the skull volume. The future work will be exploiting the sparsity of the skull data to save computation and reduce the memory requirements, so that using larger patches without increasing the GPU memory in training is possible[1].

[1]The optimal situation is that the network can consume the entire skull volume

Bibliography

Angelo, Luca et al. (Feb. 2019). "A Robust and Automatic Method for the Best Symmetry Plane Detection of Craniofacial Skeletons." In: *Symmetry* 11, p. 245. DOI: 10.3390/sym11020245 (cit. on p. 5).

Chen, Xiaojun et al. (2017). "Computer-aided implant design for the restoration of cranial defects." In: *Scientific Reports* (cit. on p. 5).

Dai, Angela, Charles Ruizhongtai Qi, and Matthias Nießner (2016). "Shape Completion Using 3D-Encoder-Predictor CNNs and Shape Synthesis." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6545–6554 (cit. on p. 6).

Dou, Qi et al. (2017). "3D deeply supervised network for automated segmentation of volumetric medical images." In: *Medical Image Analysis* 41, pp. 40–54 (cit. on p. 13).

Egger, Jan et al. (Mar. 2017). "Interactive reconstructions of cranial 3D implants under MeVisLab as an alternative to commercial planning software." In: *PLoS ONE* 12, p. 20. DOI: 10.1371/journal.pone.0172694 (cit. on p. 5).

Gall, Markus, Xing Li, et al. (2016). "Computer-aided planning and reconstruction of cranial 3D implants." In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1179–1183 (cit. on p. 5).

Gall, Markus, Alois Tax, et al. (Mar. 2019). "Cranial Defect Datasets." In: DOI: 10.6084/m9.figshare.4659565.v6. URL: https://figshare.com/articles/Cranial_Defect_Datasets/4659565 (cit. on p. 8).

Han, Xiaoguang et al. (Oct. 2017). "High-Resolution Shape Completion Using Deep Neural Networks for Global Structure and Local Geometry Inference." In: DOI: 10.1109/ICCV.2017.19 (cit. on p. 6).

Heinrich, Mattias P., Ozan Oktay, and Nassim Bouteldja (2019). "OBELISK-Net: Fewer layers to solve 3D multi-organ segmentation with sparse deformable convolutions." In: *Medical Image Analysis* 54, pp. 1–9 (cit. on p. 13).

Kamnitsas, Konstantinos et al. (2017). "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation." In: *Medical Image Analysis* 36, pp. 61–78 (cit. on p. 13).

Kazhdan, Michael M., Matthew Bolitho, and Hugues Hoppe (2006). "Poisson surface reconstruction." In: *Symposium on Geometry Processing* (cit. on p. 5).

Kazhdan, Michael M. and Hugues Hoppe (2013). "Screened poisson surface reconstruction." In: *ACM Trans. Graph.* 32, 29:1–29:13 (cit. on p. 5).

Li, Dongping et al. (2017). "Shape Completion from a Single RGBD Image." In: *IEEE Transactions on Visualization and Computer Graphics* 23, pp. 1809–1822 (cit. on p. 6).

Li, Jiaxin, Ben M. Chen, and Gim Hee Lee (2018). "SO-Net: Self-Organizing Network for Point Cloud Analysis." In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9397–9406 (cit. on p. 6).

Li, Xiaomeng et al. (2018). "3D multi-scale FCN with random modality voxel dropout learning for Intervertebral Disc Localization and Segmentation from Multi-modality MR Images." In: *Medical Image Analysis* 45, pp. 41–54 (cit. on p. 13).

Li, Yangyan et al. (2018). "PointCNN: Convolution On X-Transformed Points." In: *NeurIPS* (cit. on p. 6).

Litany, Or et al. (2017). "Deformable Shape Completion with Graph Convolutional Autoencoders." In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1886–1895 (cit. on p. 6).

Liu, Xingyu, Mengyuan Yan, and Jeannette Bohg (2019). "MeteorNet: Deep Learning on Dynamic 3D Point Cloud Sequences." In: *ArXiv* abs/1910.09165 (cit. on p. 6).

Marzola, Antonio et al. (May 2019). "A Semi-Automatic Hybrid Approach for Defective Skulls Reconstruction." In: *Computer-Aided Design and Applications* 17, pp. 190–204. DOI: 10.14733/cadaps.2020.190-204 (cit. on p. 5).

McInnes, Leland and John Healy (2018). "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction." In: *ArXiv* abs/1802.03426 (cit. on p. 29).

Mitra, Niloy Jyoti, Leonidas J. Guibas, and Mark Pauly (2006). "Partial and approximate symmetry detection for 3D geometry." In: *ACM Trans. Graph.* 25, pp. 560–568 (cit. on p. 5).

Morais, Ana (2018). "Automated Computer-aided Design of Cranial Implants-A Deep Learning Approach." MA thesis. Universidade do Minho, pp. 1–98 (cit. on p. 5).

Morais, Ana, Jan Egger, and Victor Alves (Apr. 2019). "Automated Computer-aided Design of Cranial Implants Using a Deep Volumetric Convolutional Denoising Autoencoder." In: pp. 151–160. ISBN: 978-94-024-1613-8. DOI: 10.1007/978-3-030-16187-3_15 (cit. on p. 5).

Ngo, Hanh T.-M. and Won-Sook Lee (2011). "Feature-First Hole Filling Strategy for 3D Meshes." In: *ICCV 2011* (cit. on p. 5).

Qi, Charles Ruizhongtai, Hao Su, et al. (2016). "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation." In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85 (cit. on p. 6).

Qi, Charles Ruizhongtai, Li Yi, et al. (2017). "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space." In: *NIPS* (cit. on p. 6).

Sakr, Nahla M. et al. (2018). "An effective method for hole filling in 3D triangular meshes." In: *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 1–7 (cit. on p. 5).

Schiebener, David et al. (2016). "Heuristic 3D object shape completion based on symmetry and scene context." In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 74–81 (cit. on p. 5).

Stutz, David and Andreas Geiger (2018). "Learning 3D Shape Completion from Laser Scan Data with Weak Supervision." In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1955–1964 (cit. on p. 6).

Sung, Minhyuk et al. (2015). "Data-driven structural priors for shape completion." In: *ACM Trans. Graph.* 34, 175:1–175:11 (cit. on p. 5).

Wang, Jianing, Jack H. Noble, and Benoit M. Dawant (2019). "Metal artifact reduction for the segmentation of the intra cochlear anatomy in CT images of the ear with 3D-conditional GANs." In: *Medical image analysis* 58, p. 101553 (cit. on p. 12).

Wu, Zhirong et al. (2014). "3D ShapeNets: A deep representation for volumetric shapes." In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920 (cit. on p. 6).

Yang, Xin et al. (2017). "Hybrid Loss Guided Convolutional Networks for Whole Heart Parsing." In: *STACOM@MICCAI* (cit. on p. 12).

Yang, Yaoqing et al. (2017). "FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation." In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 206–215 (cit. on p. 6).

Zhao, Wei, Shuming Gao, and Hongwei Lin (2007). "A Robust Hole-Filling Algorithm for Triangular Mesh." In: *CAD/Graphics* (cit. on p. 5).