Cranial Implant Prediction by Learning an Ensemble of Slice-Based Skull Completion Networks

Bokai Yang, Ke Fang, and Xingyu Li(✉)

University of Alberta, Edmonton, AB T6G 2R3, Canada
{bokai5,kfang1,xingyu}@ualberta.ca

Abstract. The development of automatic skull reconstruction methods has dramatically reduced the time and expense to repair skull defects. In this study, an ensemble-learning-based method is proposed for skull implant prediction. To overcome the potential overfit problem in 3-D volume analysis using deep learning, a set of 2-D defective skull images is generated by slicing 3-D volumes along the X, Y, and Z axes. We further introduce an RNN model in this method to compensate for the loss of global skull information in the 2-D implant prediction. Over the implant estimation problem in Task 1 of the AutoImplant 2021 challenge, we observe a considerable performance boost from our averaging ensemble strategy and noise removal filtering. The codes for our method as well as our pretrained models is accessible with https://github.com/YouJianFengXue/Cranial-implant-prediction-by-learning-an-ensemble-of-slice-based-skull-completion-networks.

Keywords: Cranial implant design · Deep learning · AutoImplant

1 Introduction

Defects on cranial bones are usually caused by physical damage or pathological damage to the skulls. Cranioplasty is reconstructive surgery for such skull injury repair. Traditionally, doctors put universal covers on the defective region. However, this solution results in poor aesthetic outcomes and the gap between a skull and implant may not be fully recovered [12]. Later, customized implants were designed to improve the overall cranioplasty outcomes. Patient-specific skull implant customization is a complex procedure with a relatively long waiting time and requires a dedicated CAD software [1,2,7,10]. Recently, there has been an increasing interest in artificial patient-specific implants (PSI). PSI uses computer-aided algorithms and machine learning to generate skull implants based on medical imaging of skull defects and is expected to reduce overall patient risk and surgery time in the operating room [5]. A typical example of PSI is the AutoImplant 2020 challenge, where challenging participants present various data-driven solutions based on triplet of cranial defects and corresponding skull implants. For instance,

J. Li and J. Egger (Eds.): AutoImplant 2021, LNCS 13123, pp. 95–104, 2021.
https://doi.org/10.1007/978-3-030-92652-6_8

classical statistical models such as the statistical shape model (SSM) [8] were used to estimate the skull shape for implant design. We notice that deep learning is still the major technique adopted in this challenge. Specifically, Generative Adversarial Networks (GAN) [8], Variational Autoencoders (VAE) [13], U-Net [3,4,6,9] and its variants, are the most popular generative models for skull completion and implant estimation.

For the purpose of PSI, a defective skull is usually scanned into a 3-D skull volume for downstream analysis. Intuitively, a 3-D U-Net is the candidate network to process the data volume for defective skull recovery. However, this network has many trainable parameters that require a large data set for model training. For one thing, collecting an extensive skull defect data set is expensive. For another, processing a batch of 3-D volumes involves data-intensive computation, which challenges computing resources, especially the memory of a graphic card. We present a hardware-friendly solution to skull implant prediction to address the above issues. Notably, we mitigate the data scarcity issue in model training by slicing the skull volumes into 2-D planes along X, Y, and Z axes for 2-D implant prediction. Simultaneously, the 2-D data analysis and subsequent model ensemble help to reduce the demand on computing hardware.

2 Methodology

2.1 Dataset

AutoImplant 2021 Challenge is an update of the AutoImplant 2020 Challenge. Particularly for Task 1: cranial implant design for diverse synthetic defects on aligned skulls, 570 cases that are distributed into 5 folders depending on defects' locations are available for training and 100 samples in total (i.e. 20 in each folder) for evaluation. For each training case, a triplet of defective skulls, corresponding complete skull and implant are provided. All data samples are represented in binary $512 \times 512 \times 512$ volumes and saved in NRRD format.

2.2 Motivation

3-D skull volume analysis is challenging. Training a 3-D deep model such as the 3-D U-Net on limited samples is prone to overfit, harming models' generalizability on unseen data. In the AutoImplant 2020 Challenge, Shi et al. [11] present a multi-axis slicing solution to address the issue. The method first exploits a 2D CNN network for skull implant estimation on each 2-D plane. Then the obtained skull implant slices are combined to form the final 3-D implant. This algorithm greatly mitigates the requirement of the number of data and computing resources. However, it completely abandons the global information on a skull volume in skull implant perdition. We argue that such global information, especially the continuity between adjacent skull slices, is vital for cranial defect recovery, and considering it in implant design would improve the final results. In this regard, we design an LSTM model to account for the continuity between

skull slices; furthermore, we adopt ensemble learning to fuse the outcome of our RNN model and the CNN multiaxial slice network proposed by Shi et al. [11] for final estimation.

2.3 Architecture

Figure 1 depicts the diagram of our networks. Given a 3-D defective skull volume, we generate three sets of 2-D planes along the X, Y, and Z axes, respectively. For each slice set, we train two 2-D networks for implant prediction. The CNN model estimates the implant from a single slice, and the RNN neural network takes five continuous slices as its input instead. Before the synthesis step, the system generates six 3-D implant volumes from the parallel processing of the three sets of skull slices. Finally, we combine all six outcomes together by an averaging ensemble strategy. In addition, we design two computational-efficient filters to remove isolated noise for the final output.

Specifically, to prepare 2-D images for downstream CNN and RNN models, a skull volume sample is sliced along X, Y, and Z axis. To decrease the training complexity, we remove the blank slices from the 2-D training sets. Here, blank slices are defined as the images that don't contain skull defect region or cranial bones.

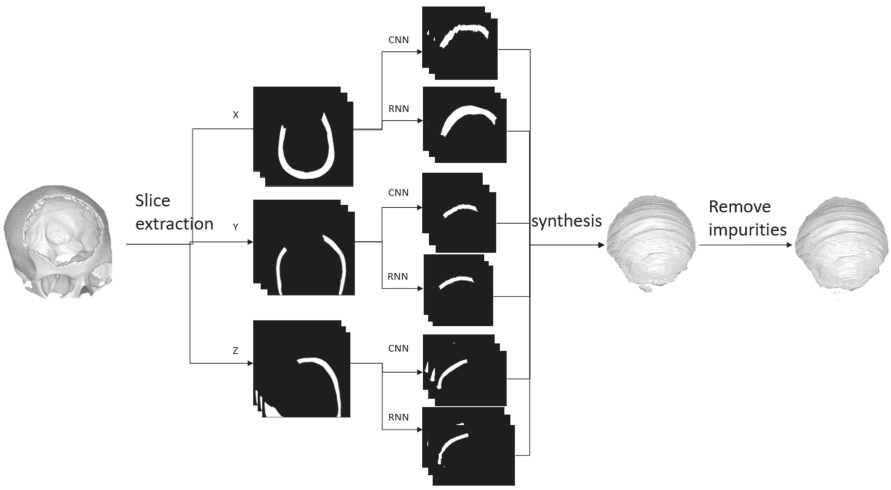

Fig. 1. Diagram of our skull implant estimation model. Given a 3-D defective skull volume, we follow the multiaxial slice network proposed by Shi et al. [11] and slice the volumes into 3 sets of 2-D planes. We design an ensemble solution that fuses the CNN and RNN outcomes for final implant prediction. The specific neural network architectures of the CNN and RNN models as well as the data flow in our RNN model are presented in Fig. 2 and Fig. 3, respectively.

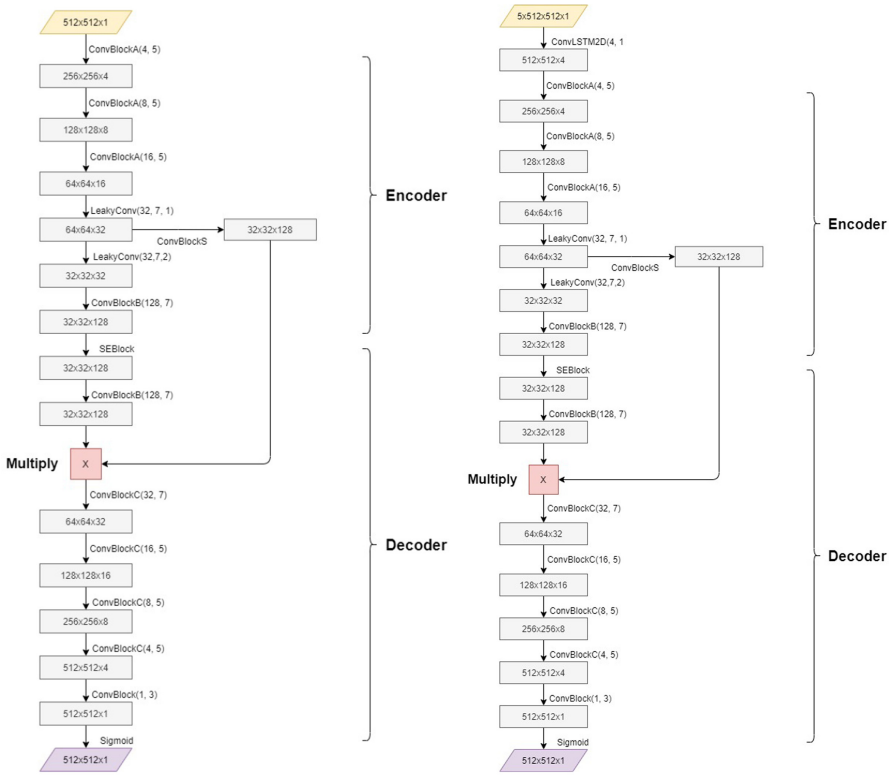

Fig. 2. (a) The architecture of our 2D CNN network and (b) The architecture of our 2D RNN model.

Network Architecture and Training. Our solution comprises two deep learning models: a CNN network that focuses on the processing of local information within one slice and an RNN model that takes advantage of continuity information among adjacent slices for skull implant prediction. Both CNN and RNN networks are composed of an encoder and a decoder, as shown in Fig. 2. The encoder projects the 2-D slices into a low-dimensional feature space, and the decoder predicts the skull implant accordingly.

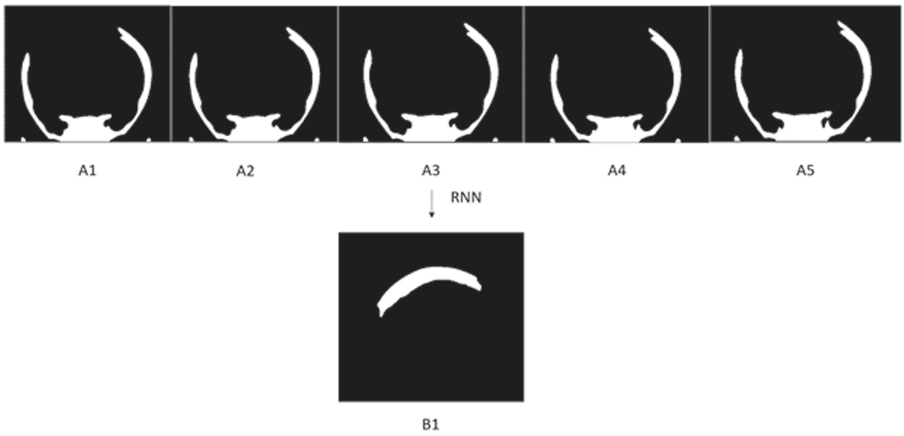

Fig. 3. Data flow of the RNN model. A1 to A5 represent the 5 continuous skull slices fed to the RNN model. The slice represented by B1 is the output of the RNN and is treated as the correspond implant of A3 in this study.

Specifically, our CNN model follows the design of the multi-axis study. The RNN model in this study adopts an LSTM module as the first layer above the CNN net, targetting to address the continuity information between adjacent slices for implant prediction. As demonstrated in Fig. 3, a 2-D skull implant is estimated based on five consecutive 2-D slices in the RNN model. To train both networks, we cast the 2-D skull implant estimation problem into a 2-D segmentation problem, where the skull defect is treated as the targeted segmentation region in a skull slice. Therefore, following the conventional segmentation setting, the DICE loss is taken as the objective function for model optimization.

$$Dice\ loss = 1 - \frac{2\Sigma_i \left| P_i * G_i \right| + \xi}{\Sigma_i \left(P_i \right)^2 + \Sigma_j \left(G_j \right)^2 + \xi}, \tag{1}$$

where P_i and G_i represent the implant prediction and corresponding ground truth at the pixel i, respectively and ξ is a smooth factor to prevent the gradient vanishing or explosion. In this study, we set $\xi = 10^{-6}$.

When we train the models, we use the Adam optimizer with a learning rate of 0.00005 and a clipnorm of 1.0. Due to the limitation of the computing resource, we set the batch size to be 1 during training.

Implant Synthesis. With the RNN and CNN models, we obtain six volumes of skull implants, each consisting of all 2-D skull estimations along either X, Y, or Z axes. Then we utilize an ensemble learning strategy to synthesize the final skull implant from the six candidates. Specifically, in both RNN and CNN models, each point in the volume is associated with a probability value indicating the likelihood of a point belonging to the implant. We compare the sum of the six likelihood values to a predefined threshold to determine if the point

contributes to the final implant synthesis. In the solution proposed by Shi et al. [11], since only three coarse skull implants are generated, 1.5 (i.e., 0.5×3) is taken as the threshold to differentiate the defective region and cranial bone in a 3D volume. However, we found this threshold inappropriate for our problem, and the resulting skull estimations had many holes and debris. This problem is especially severe when the defect is located on the front portion of the skull. We present a typical example in Fig. 4(a) with a threshold of 1.5 in our ensemble learning. To address this problem, we tried different values for the averaging threshold and discovered that 1.0 is the optimal value in our method (e.g., Fig. 4(c)). If the threshold is smaller than 1.0, the implant prediction has large, noisy parties, as shown in Fig. 4(b)).

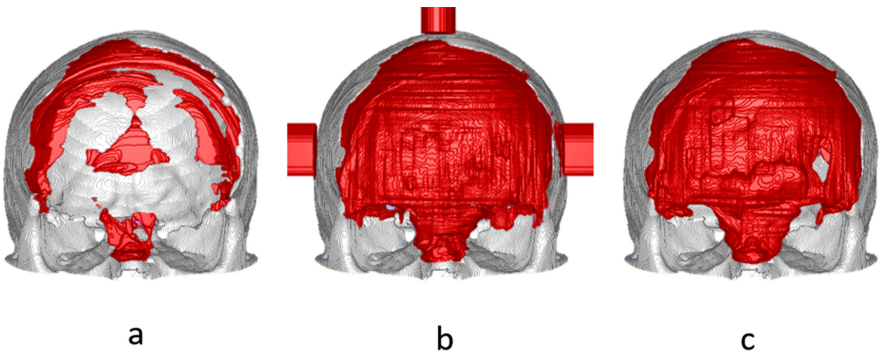

Fig. 4. Different threshold versus skull synthesis from the six coarse defect volumes. (a) Threshold of 1.5 (b) threshold of 0.5, and (c) threshold of 1.0. The grey regions correspond to defective skull and the red ones represent skull implants generated under different thresholds.

Impurity Filter. In skull synthesis, we observe small, isolated artifacts outside of skulls. Therefore, we design two simple filters to remove the isolated noise in the predicted implants, one in the dimension of $7 \times 7 \times 7$ and the other in $11 \times 11 \times 11$. Please refer to the Experiment and Result section for the qualitative and quantitative evaluation of the ensemble strategy and impurity filters.

3 Experimentation and Results

After training our networks using provided training samples, we submit the predicted implants to the challenge organizer and get feedback on the quantitative evaluation. In this problem, 3 metrics including Hausdorff distance (HD) and dice similarity score (DSC), and broader dice are used for performance assessment.

Since our method has extensive overlap with the solution proposed by Shi et al. [11], we downloaded their Github code, run their model on the Task 1 data in this challenge from scratch, and took it as our comparison baseline in this study. The statistics over the test set are presented in Table 1, where all numerical metrics are averaged over the five folders. The specific quantitative evaluation results over the five folders are presented in the Appendix. As we explained and reported in the Appendix, our submission of the baseline model had unexpected errors on the test samples in the folder of random2. So we report two sets of results for the baseline model, where the numerical values associated with "'Baseline" in the first row are computed from the results of the first four folders in Table 2 and the values in the second row marked as "'Baseline*" are averaged over evaluations across all 5 folders for your reference. The results in Table 1 suggest that both our ensemble learning strategy and the impurity filters improve the implant prediction performance. To visualize the performance boost obtained by our ensemble strategy, we present two skull implant predictions in Fig. 5 for comparison. Skull implants in the first row are predicted by the baseline CNN network only, and the examples in the second row are generated after our averaging ensemble strategy. From the figure, the implants in the first row are incomplete. After our ensemble strategy to combine CNN and RNN results, the implants in the second two are complete with more smooth surfaces.

Table 1. The comparison of DICE, border DICE, and HD95 among the baseline model, ensemble learning model, and our final solutions.

	HD95	DICE	Border DICE
Baseline	3.09	0.77	0.81
Baseline*	16.74	0.64	0.68
CNN+RNN	7.47	0.80	0.85
CNN+RNN+Impurity filter	3.33	0.81	0.86

In the future, we would like to improve this method in the following two directions. First, this method includes two simple filters to remove small unwanted parties from the synthesis implant. However, the two filters are incapable of eliminating large impurities. We want to utilize connected component analysis to remove isolated noise. Second, this study explores RNN quite naively (i.e., adding an LSTM layer on the top of a CNN model). We believe that training an entirely new RNN model from scratch will improve the overall performance.

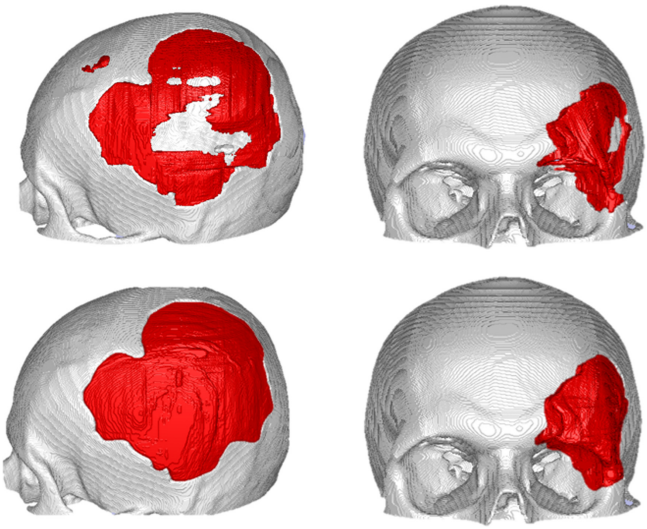

Fig. 5. This figure shows the effect of ensemble learning. Skull implants in the first row are predicted by the CNN network only and the examples in the second row are generated by our method. The implants in the first row have holes on the surface. After our ensemble strategy to combine CNN and RNN results together, the implants in the second row are both improved with more smooth and complete surfaces.

4 Conclusions

This paper proposed a new skull implant design method by inpainting defective regions in 2-D skull slices. Since the networks take 2-D images/planes as the input, the models had fewer trainable parameters and thus mitigated the negative effects of a limited number of training samples. The quantitative and qualitative results indicated that our averaging ensemble strategy over coarse implants and the two purity filters helped improve the performance.

Appendix

In this challenge, we made several submissions for algorithm assessment and improvement. The specific quantitative evaluation of our submissions that comes from the challenge organizer are presented in this section. Specifically, Table 2 reports the performance of the baseline CNN model on the test samples in the five folders. Note that our submission encountered unexpected errors over the test samples in the folder of random2. So we particularly include this information here for your reference or any further research and report the performance of the baseline through two sets of numerical values in Table 1, where the values in the first row are computed from the results of the first four folders and the second line corresponds to the performance assessment over all 5 folders. We

believe that quantitative measurement in the first row of Table 1 reflects the the performance of our baseline model. Similarly, Table 3 and Table 4 report the specific numerical results for our later submissions. Slightly different from the baseline model, the final results presented in Table 1 are averaged over the five folders.

Table 2. The quantitative results of the baseline network from the challenge organizer. Note that our submission had unexpected errors on the test samples in the folder of random2. So we faithfully mark the error here with the star sign $*$ for your reference.

	Bilateral	Frontoorbital	Parietotemporal	Random1	Random2
DICE	0.74	0.76	0.80	0.79	0.11*
Border DICE	0.80	0.79	0.86	0.82	0.10*
HD95	4.06	2.84	2.44	3.04	71.32*

Table 3. The quantitative results of the our CNN+RNN models from the challenge organizer.

	Bilateral	Frontoorbital	Parietotemporal	Random1	Random2
DICE	0.79	0.80	0.83	0.80	0.79
Border DICE	0.84	0.82	0.88	0.84	0.84
HD95	10.56	8.70	2.64	7.60	7.85

Table 4. The quantitative evaluation of the entire solution from the challenge organizer.

	Bilateral	Frontoorbital	Parietotemporal	Random1	Random2
DICE	0.80	0.81	0.84	0.80	0.80
Border DICE	0.85	0.83	0.89	0.86	0.85
HD95	3.87	2.70	2.63	3.87	3.60

References

1. Chen, X., Xu, L., Li, X., Egger, J.: Computer-aided implant design for the restoration of cranial defects. Sci. Rep. **7**, 4199 (2017)
2. Dean, D., Min, K.-J.: Computer aided design of cranial implants using deformable templates (2003)
3. Ellis, D.G., Aizenberg, M.R.: Deep learning using augmentation via registration: 1st place solution to the AutoImplant 2020 challenge. In: Li, J., Egger, J. (eds.) AutoImplant 2020. LNCS, vol. 12439, pp. 47–55. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64327-0_6

4. Ellis, D.G., Aizenberg, M.R.: Deep learning using augmentation via registration: 1st place solution to the AutoImplant 2020 challenge. In: Li, J., Egger, J. (eds.) AutoImplant 2020. LNCS, vol. 12439, pp. 47–55. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64327-0_6
5. Li, J., et al.: Towards the automatization of cranial implant design in cranioplasty: 2nd MICCAI Challenge on Automatic Cranial Implant Design, March 2021. https://doi.org/10.1007/978-3-030-64327-0
6. Mainprize, J.G., Fishman, Z., Hardisty, M.R.: Shape completion by U-Net: an approach to the AutoImplant MICCAI cranial implant design challenge. In: Li, J., Egger, J. (eds.) AutoImplant 2020. LNCS, vol. 12439, pp. 65–76. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64327-0_8
7. Ming-Yih, L., Chong-Ching, C., Chao-Chun, L., Lun-Jou, L., Yu-Ray, C.: Custom implant design for patients with cranial defects. IEEE Eng. Med. Biol. Mag. **21**, 38–44 (2002)
8. Pimentel, P., et al.: Automated virtual reconstruction of large skull defects using statistical shape models and generative adversarial networks. In: Li, J., Egger, J. (eds.) AutoImplant 2020. LNCS, vol. 12439, pp. 16–27. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64327-0_3
9. Eder, M., Li, J., Egger, J.: Learning volumetric shape super-resolution for cranial implant design. In: Li, J., Egger, J. (eds.) AutoImplant 2020. LNCS, vol. 12439, pp. 104–113. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64327-0_12
10. Scharver, C., Evenhouse, R., Johnson, A., Leigh, J.: Pre-surgical cranial implant design using the Paris/SPL trade/prototype. IEEE Virtual Real. **2004**, 199–291 (2004)
11. Shi, H., Chen, X.: Cranial implant design through multiaxial slice inpainting using deep learning. In: Li, J., Egger, J. (eds.) AutoImplant 2020. LNCS, vol. 12439, pp. 28–36. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64327-0_4
12. Gord von Campe and Karin Pistracher. Patient specific implants (PSI) cranioplasty in the neurosurgical clinical routine. In: AutoImplant 2020, LNCS 12439, pp. 1–9, 2020 (2020)
13. Wang, B., et al.: Cranial implant design using a deep learning method with anatomical regularization. In: Li, J., Egger, J. (eds.) AutoImplant 2020. LNCS, vol. 12439, pp. 85–93. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64327-0_10