CAMAlyzer: A 3DSlicer extension for AI segmentation and generation of proximal femur 3D models

Benjamin A. Rodríguez
Faculty of Engineering and Sciences
Adolfo Ibáñez University
Viña del mar, Chile
benjamirodriguez@alumnos.uai.cl

Alejandro A. Espinoza
Dept. of Diagnostic Radiology and Nuclear Medicine
Rush University Medical Center
Chicago, Illinois, USA
ORCID 0000-0002-3792-515X

Shane Nho
Dept. of Orthopedic Surgery
Rush University Medical Center
Chicago, Illinois, USA
Shane.Nho@rushortho.com

Miguel Carrasco
School of Informatics and Telecomunications
Diego Portales University
Santiago, Chile
miguel.carrasco@mail.udp.cl

Juan F. Vivanco
Faculty of Engineering and Sciences
Adolfo Ibáñez University
Viña del mar, Chile
juan.vivanco@uai.cl

***Abstract*—Femoroacetabular impingement syndrome (FAIS) is a condition that implies increased intra-articular forces due to abnormal morphology, leading to pain, reduction in the range of motion, and even early development of hip osteoarthritis [1]. Diagnosing FAIS is challenging, and there is a lack of tools that implement the state-of-the-art techniques based on the process of 3D modeling, considering that these are time-consuming, require a large amount of data, and user input [2]–[4].A 3D Slicer extension was developed, implementing a Machine learning pipeline that allows for generating a 3D reconstruction of the proximal femur in 16.2 ± 1.9 seconds. This reconstruction has an error close to other state-of-the-art techniques with a 95% Hausdorff distance of 4.8 ± 2.1 $[mm]$. This pipeline starts with an automatic Deep learning based segmentation using a 3D U-net architecture that was trained with a set of 3T Flash DIXON MRI images (27 images with 208 slices, 256x256 px, 0.7 mm thickness) of patients diagnosed with FAIS, which performed with 0.91 and 0.82 for the DICE index. This segmentation was later improved using morphological operations and the DBSCAN algorithm. This work allows us to obtain a 3D model that is faithful to the patient's proximal femur, reducing segmentation time without losing the accuracy of manual segmentation. Finally, the extension it's available for downloading and use at this link: https://github.com/VenjaminRodriguezR/CAMAlyzer**

***Index Terms*—Semantic segmentation, medical imaging processing, Machine Learning, Deep Learning**

I. Introduction

Hip osteoarthritis can result from increased intra-articular forces due to abnormal morphology, excessive joint loading, or a combination of both [1]. In particular, femoroacetabular impingement syndrome (FAIS) creates an abnormal morphology at the junction between the femoral head and the acetabulum, potentially leading to severe labral and/or cartilage damage, and even early development of osteoarthritis, significantly impacting patients' quality of life. Diagnosing FAIS is challenging, as traditional methods, such as measuring α or $LCEA$ angle, struggle to capture the three-dimensional (3D) phenotypes in patients. Current techniques, such as those based on 3D modeling, are time-consuming and require expert input [2].

Therefore, there is a lack of accessible and effective tools for accurate 3D diagnosis and potentially unnecessary examinations. The quantification of abnormal morphologies using a specific healthy bone reference has been developed [2]; however, these methods face challenges due to the extensive time required for completion (4-6 hours). Segmentation is usually the bottleneck in any image-based 3D model biomechanics analysis.

Although automated tools, including those based on Deep Learning or Statistical Shape Modeling, have been proposed to expedite these time-consuming processes [3], [4], they are not yet widely available for research or clinical use. In this study, a Machine Learning Pipeline was implemented as a 3D Slicer extension, making it accessible to everyone. This extension aims to accelerate the process of obtaining a 3D model of the proximal femur as soon as possible, while maintaining the same quality as manual segmentation.

II. Materials and Methods

A. Patient selection and data collection

This study was conducted in accordance with the ethical approval obtained from the Rush University Medical Center Institutional Review Board (IRB) under protocol number ORA#18121304. All participants provided written informed consent prior to enrollment and the acquisition of MRI data. The consent procedure included explicit authorization for the use of anonymized imaging data for research purposes. No identifying information was retained, and all data were processed in de-identified form to ensure patient confidentiality.

For this study, MRI data from Rush University Medical Center was recorded with a 3T magnetic resonance scanner (Siemens Magnetom Verio. Siemens, Erlangen, Germany) using a 3D axial T1 VIBE DIXON, fat saturated sequence. Patients from both genders aged 18-45 with a known diagnostic of FAI were included in the study as part of their presurgical workup. Since 3T scanners have a stronger magnetic field than the more commonly available 1.5T units, their use enables new sequences that can display the contrast between soft and bony tissue, as well as highlight tissue conditions. It has been shown that this type of scan is a viable and less harmful option for the patient when seeking to compare 3D models to quantify femoroacetabular impingement [2].

Once the images were captured, they were exported and stored in DICOM format. This type of file can be opened with any medical image visualization software. For the MR images, the region of interest (ROI) was selected, that is, the area where the suspected malformation is present. The data set consisted of 3T Flash DIXON MRI images (27 images with 208 slices, 256x256 px, $0.7mm$ thickness) of the hip joint of patients previously diagnosed with FAI. Water-phase images were used to enhance bone visualization.

B. Manual segmentation

The ground-truth segmentations were created using the 3D Slicer segment editor [5] and the MONAI Label Framework [6]. The segmentations was performed by Masters Student in Biomedical Engineering, combining semiautomatic and manual tools. These two tools were available in 3D Slicer: the paint and erase effects, and the "grow from seeds" effect. Initially, all pixels were set to 0. The femur region was manually labeled with the paint tool, assigning a value of 1 to the selected pixels. Similarly, background pixels were marked for erasure, creating a binary image. The "grow from seeds" operation was then applied, using an algorithm based on the method by [7]. Finally, background segmentation was removed, and the femur segmentation was cleaned using the erase tool. Using MONAI Label, both the segmentation and the original volume were automatically saved as separate NIfTI files, preserving the original dimensions.

C. Network architecture

3D U-net [8] was designed specifically for volumetric segmentation that learns from sparsely annotated volumetric images. This network takes 3D volumes as input and processes them with the corresponding 3D operations, in particular, 3D convolutions, 3D pooling, and inverse 3D convolution. The standard u-net, it has an analysis and a synthesis path, each with four resolution steps [8]. In the analysis path, each layer contains two $3\times3\times3$ convolutions, each followed by a rectified linear unit (ReLu), and then a $2\times2\times2$ max pooling with strides of two in each dimension [8]. In the synthesis path, each layer consists of an upconvolution of $2 \times 2 \times 2$ by strides of two in each dimension, followed by two $3 \times 3 \times 3$ convolutions, each followed by a ReLu [8]. Shortcut connections from layers of equal resolution in the analysis path provide the essential high-resolution features to the synthesis path [8]. In the last layer, a $1\times1\times1$ convolution reduces the number of output channels to the number of labels, which is 3 in our case [8]. This standard architecture has $19.069.955$ parameters in total [8].

D. 3D U-net optimization

A 3D-Unet [8] base model was optimized using the Optuna framework [9]. The hyperparameters to optimize were a specific selection from the proposed in [10].

A range of 3D U-Net variants was evaluated by varying four key hyperparameters: dropout rate, optimizer type, batch size, and convolutional kernel size. Dropout was sampled continuously between 0% and 60% to assess regularization effects; optimizers included Adam, RMSprop, Adadelta and Adagrad; batch sizes of 1, 2 or 4 were tested to balance memory constraints and gradient stability; and cubic kernels of size [1x1x1], [3x3x3] or [5x5x5] were compared to study receptive field impact.

Each Optuna trial randomly reshuffled the 27 volumes into 70% training, 20% validation, and 10% testing partitions, with a unique random seed per trial to capture variability in data splits. Trials ran for up to 400 epochs, with per-epoch reporting of training loss, validation loss, and Dice coefficient, and occasional pruning every 100 epochs when validation performance plateaued. All trials were seeded to ensure reproducibility of both hyperparameter sampling (seed = 42) and per-trial data splits.

Inference during validation and testing employed a sliding-window strategy (window size [96x96x96], overlap factor of 4) to accommodate high-resolution volumes, followed by sigmoid activation and a 0.5 threshold for segmentation maps. Performance was primarily measured by the mean Dice score, with additional logging of average surface distance to capture contour accuracy.

The U-Net architecture used in all trials followed a standard 3D configuration with five encoding-decoding levels. Each level doubled the number of feature channels, using the configuration (16, 32, 64, 128, 256). The network included residual units (two per level) to improve gradient flow during training. Downsampling was performed with strides of (2, 2, 2), and the number of input and output channels was fixed at 1, as the task involved binary segmentation.

1) DICE Coefficient: The different architectures were optimized and compared in basis of the DICE coefficient, also called the overlap index, wich is the most widely used metric in the evaluation of medical segmentations [11]. It is used to measure the reproducibility of segmentation, as well as to compare actual segmentation with automatic segmentation. The value of this index is given by equation 1.

$$DICE = \frac{2TP}{2TP + FP + FN} \tag{1}$$

E. Machine Learning Pipeline

The evaluation was performed on 11 test cases. For each case, three surface models were generated: the raw surface

obtained directly from the predicted segmentation, the surface after applying morphological opening to reduce small artifacts, and the final version after using DBSCAN to remove disconnected components and isolate the femoral head.

All surfaces were reconstructed using the Marching Cubes algorithm and converted to 3D meshes [18]. Initially, this algorithm was defined as one that creates triangle models of constant density surfaces from 3D medical data, using a divide-and-conquer approach to generate inter-slice connectivity it creates a case table that defines triangle topology [18]. This algorithm extracts an iso-surface from volumetric data by processing the image volume cube-by-cube in a divide-and-conquer fashion. Each cube, defined by eight neighboring voxels, is analyzed to determine whether its vertices lie inside or outside the segmented structure, producing an 8-bit configuration index. From a precomputed case table of 14 unique topological patterns, the algorithm interpolates edge intersections, estimates local surface normals from voxel gradients, and constructs triangles to approximate the continuous surface. Repeating this procedure across all voxel cubes yields a polygonal mesh representation of the segmented anatomy. The resulting triangulated surface enables direct visualization, geometric analysis, and rendering within conventional 3D graphics frameworks [18].

To enhance the quality of the reconstructed mesh, a Laplacian smoothing procedure was applied following surface reconstruction [19]. This method iteratively adjusts vertex positions to reduce geometric noise and irregularities while preserving the overall topology of the mesh. Each vertex is relocated toward the average position of its neighboring vertices, producing a smoother and more uniform surface. The process is controlled by a relaxation parameter that determines the degree of smoothing applied in each iteration. Boundary vertices are typically kept fixed to maintain mesh integrity, and if the mesh is constrained to a particular surface, updated vertices are reprojected onto it. Overall, Laplacian smoothing acts as a discrete diffusion process that improves mesh smoothness and visual coherence without altering its connectivity [19].

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [20] was used to clean residual noise and isolate the 3D surface of the proximal femur by clustering dense regions in point clouds without requiring a predefined number of clusters and with robustness to outliers [21]. It classified points into core, border, and noise categories, removing sparse noise and retaining the largest cluster as the main femoral surface. The resulting denoised point cloud was reconstructed into a watertight surface using Poisson surface reconstruction [22], which solves a Poisson equation to recover a smooth mesh that preserves the original geometry.

The acuracy of the models in each step were compared using a distance metric to the ground truth reconstruction. The distance was computed for each of the three stages—raw prediction, after morphological opening, and after DBSCAN—and stored per subject. Summary statistics, including mean and standard deviation, were calculated across all subjects for each method.

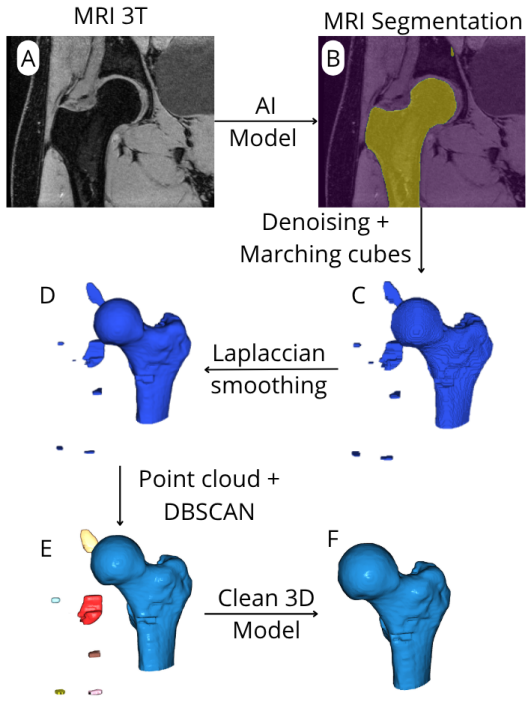

Fig. 1. Overview of the automated 3D reconstruction pipeline. (A) Original 3T MRI image. (B) AI-based segmentation of the femur. (C) 3D surface generation using denoising and Marching Cubes. (D) Laplacian smoothing. (E) Point-cloud filtering with DBSCAN to remove artifacts. (F) Final cleaned 3D femoral model.

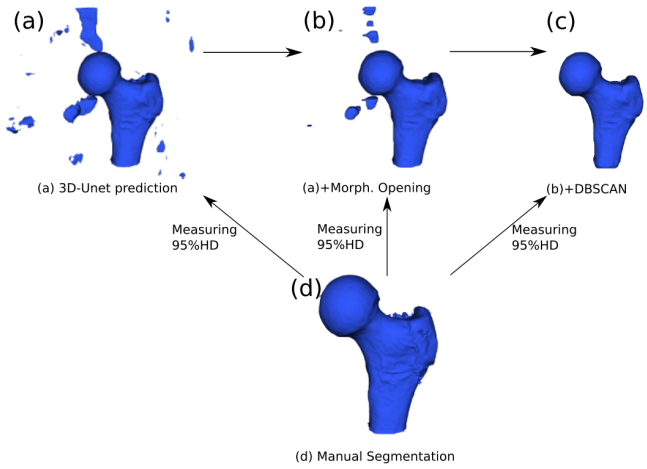

Fig. 2. Evaluation of the pipeline steps. (a) Raw 3D U-net Segmentation. (b) Automatic Segmentation and morphological operations. (c) Automatic Segmentation, morphological operations, and DBSCAN.

1) 95% Hausdorff distance: Each automated surface was compared to the corresponding manually segmented model using the 95th percentile Hausdorff distance (95% HD) as a geometric error metric. Before computing the distance, the meshes were rigidly aligned using the Iterative Closest Point (ICP) algorithm to account for possible spatial mismatches.

For each mesh pair, a point-to-point distance was computed in both directions, and the 95% HD was calculated to reflect the highest surface deviations while minimizing sensitivity to outliers.

The Hausdorff distance (HD) has been widely used to quantify the error between two surfaces [12]. The distances $d(p, S')$ between a point p from the surface S and another surface S' it's defined as:

$$d(p, S') = \min_{\substack{p \,\epsilon\, S \\ p' \,\epsilon\, S'}} ||p - p'||_2 \quad (2)$$

Where p' is the closest point to p in the surface S' and $||.||_2$ represents the norm between two points. Then, the 95% Hausdorff distance, denoted as ($95\%\text{HD}(S, S')$), is the 95$^{\text{th}}$ percentile of the set of all such distances computed from every point in (S) to (S'). Considering that $95\%\text{HD}(S, S') \neq 95\%\text{HD}(S', S)$, the convention is to use the maximum between the two, therefore:

$$95\%\text{HD}(S, S') = max\{95\%\text{HD}(S, S'), 95\%\text{HD}(S', S)\} \quad (3)$$

F. Statiscal Analysis

Due to reduced sample size (n=11), $95\%HD$ and generation time were compared between models from each pipeline step and manual segmentation using Friedman and Wilcoxon tests.

G. CAMAlyzer 3DSlicer extension

The CAMAlyzer extension it's available for downloading and use at this link: https://github.com/VenjaminRodriguezR/CAMalyzer.

III. Results

A. 3D U-net optimization

A study was conducted on a total of 40 tests with different variations of a 3D U-Net architecture. According to the results provided by the Optuna optimizer, a total of 5 parameter combinations were obtained that exceeded a DICE index of 0.8. Achieving the best results with the Adadelta optimizer, [3x3x3] kernel size, batch size of 1, and dropout of 0.1490, yielding DICE scores of 0.91 for validation and 0.82 for testing.

In general, these studies found that the best models were obtained using the Adadelta optimizer, a batch size of 1, and a kernel size of [3x3x3], varying only the dropout rate between each combination. This rate remained between 0.04 and 0.21 for the 10 best performances.

In terms of training, according to the evolution metrics in the figure 3(a), the model can learn along the validation metric. Nevertheless, some overfitting appears in the loss curves according to the difference between the train and the validation loss curves (figure 3(b))

TABLE I
BETTER COMBINATIONS OF THE STUDY

Batch size	Kernel size	Optimizer	Dropout	Validation DICE	Test DICE
1	3	Adadelta	0.1490	**0.91**	0.82
1	3	Adadelta	0.0589	0.87	0.82
1	3	Adadelta	0.2057	0.86	**0.87**
1	3	Adadelta	0.0451	0.82	0.85
1	3	Adadelta	0.1877	0.80	0.82
1	3	Adadelta	0.1397	0.79	0.82
1	3	Adadelta	0.2269	0.78	0.81
1	3	Adadelta	0.1686	0.72	0.80
1	3	Adadelta	0.1466	0.71	0.84
1	3	Adadelta	0.2065	0.69	0.83

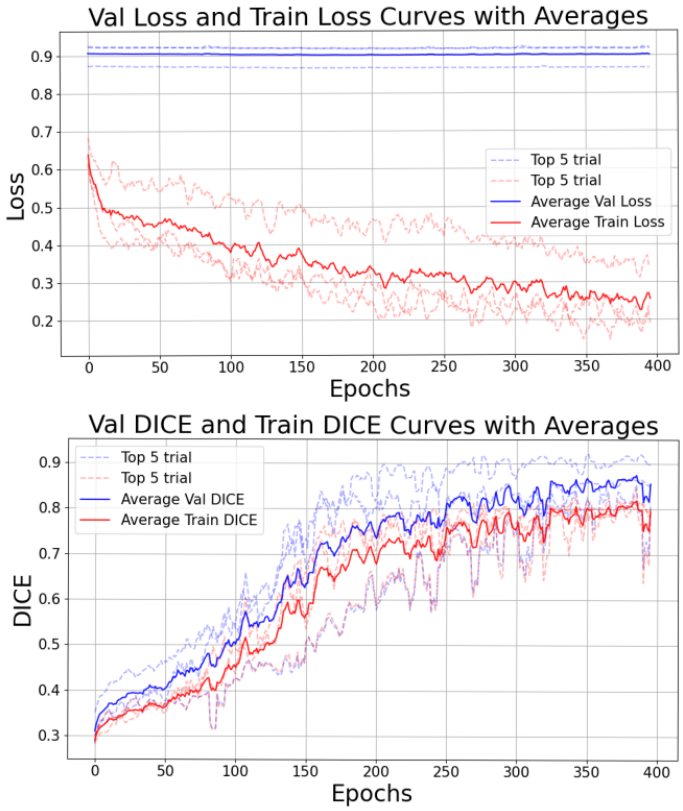

Fig. 3. Performance of the top five trials over 400 epochs for both loss (top) and DICE score (bottom). Dashed lines represent individual trials and solid lines the averages. In the loss plot, it is revealed that training loss decreases steadily while validation loss remains high and nearly constant. The DICE plot, both training and validation scores increase over time, with validation consistently outperforming training.

B. Machine Learning Pipeline

For Step A, the raw surface model was generated directly from the predicted segmentation. The average processing time was 3.6 ± 0.7 seconds. The 95th percentile Hausdorff distance (95% HD) compared to the manual ground truth was 82.2 ± 20.6 $[mm]$.

In Step B, after applying morphological opening to reduce small artifacts, the average processing time increased to 7.7 ± 0.9seconds. The 95% HD decreased to 32.4 ± 29.2 $[mm]$, presenting a larger variability.

Finally, in Step C, after using DBSCAN to remove disconnected components and isolate the femoral head, the average processing time further increased to 16.2 ± 1.9 seconds. The 95% HD was significantly reduced to 4.8 ± 2.1 $[mm]$.

These measurements reflect the geometric accuracy improvements and computational cost increases across the three steps of the pipeline. According to the statistical analysis, the results showed a significant progressive reduction ($p < 0.001$) in surface deviation, but increased significantly ($p < 0.001$) the time of computation of the process.

Figure 5 highlights that the improvement in pipeline performance occurs across all study subjects. In addition to a reduction in the method variation in each step of the pipeline, which is consistent across different subjects.

IV. Discussion

The 3D Slicer extension presented uses the Machine Learning pipeline described in Section II-E for generating a feasible reconstruction of the proximal femur for later use in 3D biomechanical analysis. This pipeline combines several techniques to optimize the reconstruction of a 3D model of the proximal femur. A deep-learning-based automatic segmentation, followed by morphological post-processing operations and the use of the DBSCAN algorithm on the point cloud version of the reconstruction to extract only the main structure.

During the training process, several hyperparameters were optimized, resulting in an optimal set of hyperparameters with a batch size of 1, a kernel size of [3x3x3], the Adadelta optimizer, and a dropout rate of 0.1490. Particularly, according to the results presented in Table I, all the best models shared the same hyperparameters, except for the dropout rate. Considering the characteristics of the dataset, the results of this optimization are congruent, as it was a small dataset (27 images). Therefore, to better learn the patterns in the data, the model required a smaller batch size. The same applies to the kernel size, since the model was trained with patches of dimension [96x96x96], as the best models performed better with a small kernel size. Although using a batch size of 1 is not recommended, a larger batch size significantly reduced performance metrics. In addition, this alternative was preferred due to limitations in data quantity and computing power.

Also, since these conditions and hyperparameters tend to produce overfitting, it makes sense that most of the values of the dropout rate were higher than 0, because this helps to counter the overfitting tendencies of the configuration. This is illustrated in the loss and metrics curves, because, even though

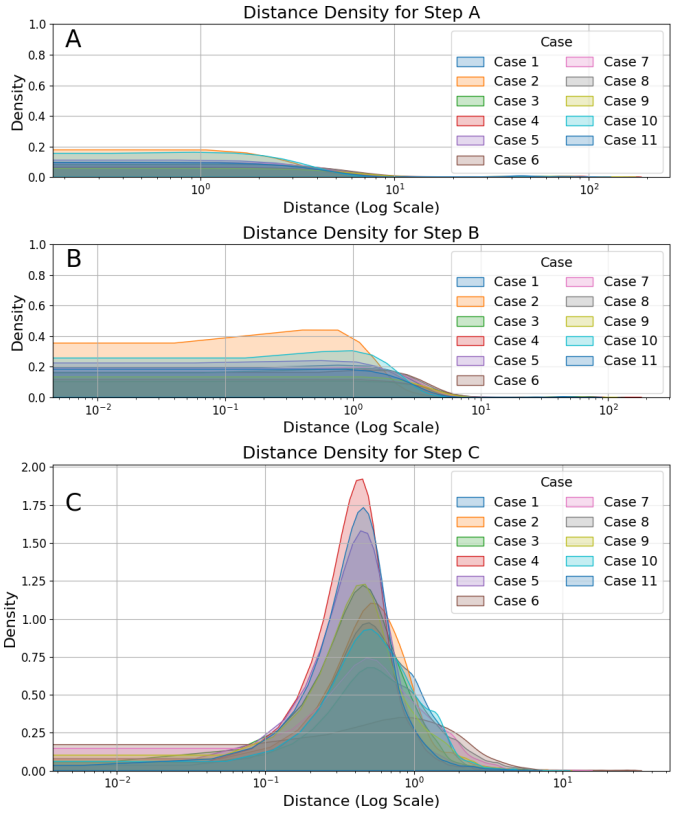

Fig. 4. Density of the distances between the automated model and ground truth model for each case study at different steps of the pipeline. In Step A, all cases show low densities with values concentrated at small distances. Step B exhibits slightly higher densities for some cases, especially Case 3, but the distributions remain mostly skewed toward smaller distances. In Step C, the density peaks sharply around distances close to 1 for all cases, indicating a more consistent distribution pattern compared to Steps A and B

it appears as an overfitting between the train and validation curve, the metric curves always tend to improve. Considering this, the post-processing proposed in the Machine Learning pipeline is essential to counteract this overtraining.

The metrics proposed in Table I prove that in general, a 3D U-net based architecture is capable of learning the key patterns needed to segment the proximal femur in a set of 3T Flash DIXON MR water-phase images. These results are comparable to those presented in [3], [13], [14], [15], where the 3D CNN architecture usually performs above 0.9 for the DICE Index.

Nevertheless, as seen in Figure 4, the best model correctly identifies the position and main shape of the proximal femur in the MRI, but presents some noise in pixels (Fig 4(c)). This means that even though the model has a good performance in the case of the True Positive pixels of the segmentation, it also presents some False Positive segmentations.

After the final reconstruction, in most cases, the error is distributed in zones that don't affect the future use of the model to quantify the CAM morphology. As shown in the figure 6, the bigger error in terms of distance between the automatic reconstruction and the ground truth reconstruction remains

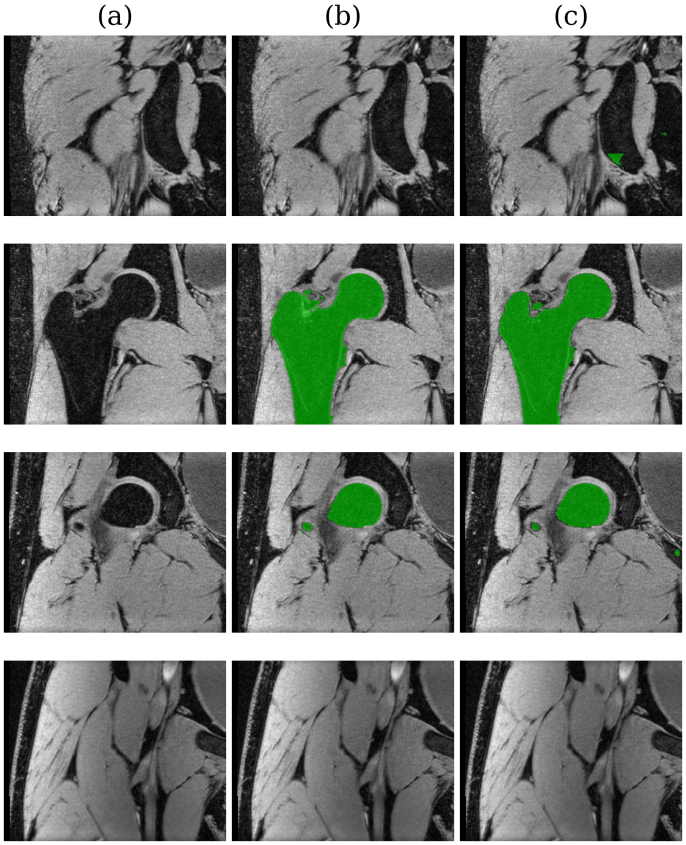

Fig. 5. MRI slices in three columns for different anatomical views: (a) the original images without annotations, (b) manual segmentations shown in green highlighting the target structures, and (c) automatic segmentations in green for comparison with the manual annotations.

closer to areas far from the usual position of the malformation (closer to the femoral neck). This has been demonstrated by [16], [17], where statistical shape modeling analyses demonstrated that, in FAIS patients, the cam deformity typically occurs at the head–neck junction of the proximal femur, away from the regions where our highest reconstruction errors were observed.

In summary, we have shown that the method proposed, which was implemented in a 3D Slicer extension, can generate an automatic reconstruction of the proximal femur from 3T Flash Dixon MRI, using a combination of Deep learning and Machine Learning algorithms. This study is not without limitations, as we note that a pretrained model is necessary to run efficiently. The same methodology presented here should work with other types of medical imaging, but with the necessity of training a separate model.

V. Conclusion

In this paper, we present a 3D Slicer extension that implements a Machine Learning algorithm to reconstruct MRI-based proximal femur models. We contribute this routine as part of a larger effort to analyze the size of the cam lesion in FAIS patients, harnessing the power of the patient-specific 3d model.

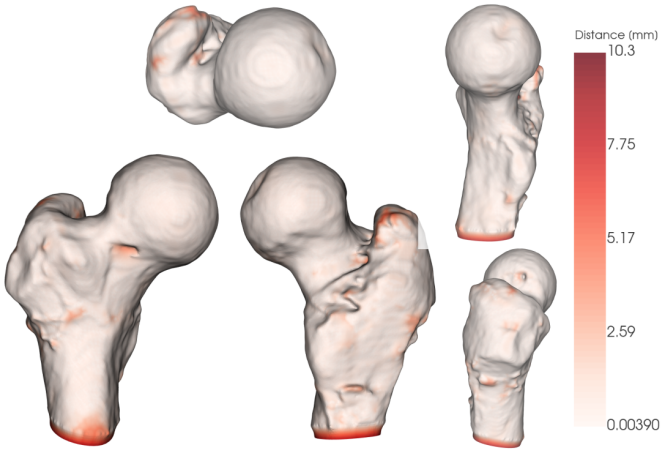

Fig. 6. Colormap of the error between the final reconstruction and the ground truth in *mm*. The bigger error areas are presented in sections far from the usual CAM deformation or in the areas where the structure has more curvature.

Acknowledgment

This work was supported in part by a Pilot grant from the Institute for Translational Medicine (NIH NCCATS UL1 TR002389), and for FONDECYT projects from the National Research and Development Agency of Chile (FONDECYT 1231708 and FONDECYT 1240197).

References

[1] Casartelli, N. C., Maffiuletti, N. A., Valenzuela, P. L., Grassi, A., Ferrari, E., van Buuren, M. M. A., Nevitt, M. C., Leunig, M., & Agricola, R. (2021). Is hip morphology a risk factor for developing hip osteoarthritis? A systematic review with meta-analysis. Osteoarthritis and cartilage, 29(9), 1252–1264. doi:10.1016/j.joca.2021.06.007

[2] Guidetti M, Malloy P, Alter TD, Newhouse AC, Nho SJ, Espinoza Orías AA. Noninvasive shape-fitting method quantifies cam morphology in femoroacetabular impingement syndrome: implications for diagnosis and surgical planning. J Orthop Res. 2023; 41: 1256-1265. doi:10.1002/jor.25469

[3] Bugeja, J. M., Xia, Y., Chandra, S. S., Murphy, N. J., Eyles, J., Spiers, L., Crozier, S., Hunter, D. J., Fripp, J., & Engstrom, C. (2022). Automated volumetric and statistical shape assessment of cam-type morphology of the femoral head-neck region from clinical 3D magnetic resonance images. Quantitative Imaging In Medicine And Surgery, 12(10), 4924-4941. doi:10.21037/qims-22-332

[4] Goparaju, A., Iyer, K., Bône, A., Hu, N., Henninger, H. B., Anderson, A. E., Durrleman, S., Jacxsens, M., Morris, A., Csecs, I., Marrouche, N., & Elhabian, S. Y. (2021b). Benchmarking off-the-shelf statistical shape modeling tools in clinical applications. Medical Image Analysis, 76, 102271. doi:10.1016/j.media.2021.102271

[5] Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J., Pujol, S., Bauer, C., Jennings, D., Fennessy, F., Sonka, M., Buatti, J., Aylward, S., Miller, J. V., Pieper, S., & Kikinis, R. (2012). 3D Slicer as an image computing platform for the Quantitative Imaging Network. Magnetic Resonance Imaging, 30(9), 1323-1341. doi:10.1016/j.mri.2012.05.001

[6] Diaz-Pinto, A., Alle, S., Nath, V., Tang, Y., Ihsani, A., Asad, M., Pérez-García, F., Mehta, P., Li, W., Flores, M., Roth, H. R., Vercauteren, T., Xu, D., Dogra, P., Ourselin, S., Feng, A., & Cardoso, M. J. (2024b). MONAI Label: A framework for AI-assisted interactive labeling of 3D medical images. Medical Image Analysis, 95, 103207. doi:10.1016/j.media.2024.103207

[7] Zhu, L., Kolesov, I., Gao, Y., Kikinis, R., & Tannenbaum, A.R. (2014). An Effective Interactive Medical Image Segmentation Method Using Fast GrowCut.

[8] Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O. (2016). 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In: Ourselin, S., Joskowicz, L., Sabuncu, M., Unal, G., Wells, W. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016. MICCAI 2016. Lecture Notes in Computer Science (), vol 9901. Springer, Cham. doi:10.1007/978-3-319-46723-8_49.

[9] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. En Proceedings 25th ACM SIGKDD international conference on knowledge discovery and data mining. doi:10.1145/3292500.3330701

[10] Hassanzadeh, T., Essam, D. & Sarker, R. Evolutionary Deep Attention Convolutional Neural Networks for 2D and 3D Medical Image Segmentation. J Digit Imaging 34, 1387–1404 (2021). doi:10.1007/s10278-021-00526-2.

[11] Taha, A. A., & Hanbury, A. (2015). Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. BMC Medical Imaging, 15(1), 29. doi:10.1186/s12880-015-0068-x

[12] Aspert, N., Santa-Cruz, D., & Ebrahimi, T. (n.d.). MESH: measuring errors between surfaces using the Hausdorff distance. Proceedings. IEEE International Conference on Multimedia and Expo. doi:10.1109/icme.2002.1035879

[13] Deniz, C.M., Xiang, S., Hallyburton, R.S. et al. Segmentation of the Proximal Femur from MR Images using Deep Convolutional Neural Networks. Sci Rep 8, 16485 (2018). doi:10.1038/s41598-018-34817-6

[14] Zeng, G., Schmaranzer, F., Degonda, C., Gerber, N., Gerber, K., Tannast, M., ... & Lerch, T. D. (2021). MRI-based 3D models of the hip joint enables radiation-free computer-assisted planning of periacetabular osteotomy for treatment of hip dysplasia using deep learning for automatic segmentation. European journal of radiology open, 8, 100303.

[15] Liu, M., Chen, Y., Tian, A., Wu, X., Shen, M., Gong, T., & Lee, J. (2025). Performance Analysis of Deep Learning Models for Femur Segmentation in MRI Scan. arXiv preprint arXiv:2504.04066.

[16] Harris, M. D., Datar, M., Whitaker, R. T., Jurrus, E. R., Peters, C. L., & Anderson, A. E. (2013). Statistical shape modeling of cam femoroacetabular impingement. Journal of Orthopaedic Research, 31(10), 1620-1626.

[17] Goparaju, A., Iyer, K., Bône, A., Hu, N., Henninger, H. B., Anderson, A. E., ... & Elhabian, S. Y. (2022). Benchmarking off-the-shelf statistical shape modeling tools in clinical applications. Medical image analysis, 76, 102271.

[18] William E. Lorensen and Harvey E. Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. SIGGRAPH Comput. Graph. 21, 4 (July 1987), 163–169. doi:10.1145/37402.37422

[19] Field, D.A. (1988) Laplacian Smoothing and Delaunay Triangulations. Communications in Numerical Methods in Engineering, 4, 709-712. doi:10.1002/cnm.1630040603.

[20] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).

[21] Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. D. F., & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. PloS One, 14(1), e0210236. doi:10.1371/journal.pone.0210236

[22] Kazhdan, M., Bolitho, M., & Hoppe, H. (2006, June). Poisson surface reconstruction. In Proceedings of the fourth Eurographics Symposium on Geometry Processing (Vol. 7, No. 4).