Deep Learning for Automated Segmentation of the Hip Joint in X-ray Images

A study of the accuracy of a ResUNet-based approach for predicting the minimum joint space width along the weight-bearing part of the hip joint in a 2D image, in comparison to BoneFinder ground-truth data

Dragoș Ileana[1]
Supervisor(s): Jesse Krijthe[1], Gijs van Tulder[1], Myrthe van den Berg[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: Dragoș Ileana

Final project course: CSE3000 Research Project
Thesis committee: Jesse Krijthe, Gijs van Tulder, Myrthe van den Berg, Xucong Zhang

Abstract

Hip osteoarthritis is a widespread disease, with medical experts facing difficulties in this illness, due to a lack of standard grading score. Nevertheless, the minimum joint space width remains the most important score for osteoarthritis severity. Manual estimation of this metric is a tedious task, which can greatly benefit from employing an automated tool. While some research has been done in developing such a tool using deep learning methods, a novel and promising approach, most lack annotated data to use for training, which can be hard to obtain. Thus, thus research aims at developing a deep learning approach towards estimating the minimum joint space, while using automatically labels produced with an existing algorithm.

1 Introduction

Hip osteoarthritis (*hip OA* or *HOA*) is a chronic disease that affects the hip joint, causing pain and stiffness [10]. The disease is characterised by tearing of the joint cartilage and bone, generally appearing with aging. Although there is no permanent cure, there are treatments that can reduce its impact such as medications and physical therapy. In later stages of the disease, a person may require a total hip replacement [10]. Early diagnosis of HOA can enable the patient to start receiving treatment at earlier stages, reducing the negative effects of the disease and helping avoid the total hip replacement surgical operation.

Hip OA is identified by means of a physical examination, but medical imaging investigations can provide further assistance by confirming the presence of HOA, sometimes even in the absence of symptoms, and allowing medical professionals to observe its development over time [10]. One key parameter used for identifying HOA and assessing HOA progression is the "joint space width" (JSW), describing the distance between the bones of the joint on the radiograph [8][4]. Currently, in the clinic, the estimation of JSW is done manually, by visually identifying the narrowest point across the joint space and measuring the distance at that point as the minimal joint space width (min-JSW or mJSW) [13]. The difficulty of analysing HOA using medical imaging stems from the lack of a standard radiological definition and grading scale for OA, resulting in significant inter- and intra-reader variability when classifying the severity of OA or measuring the JSW [10][9][8][2].

One solution to the reader variability problem is an automated tool to measure the JSW on radiographs, thanks to the expected increase in precision with this approach [4]. Some work in designing an automated tool for diagnosing hip osteoarthritis is presented in Andersen et al. [2], showcasing an algorithm for finding the minimum joint space width (mJSW) and benchmarking the performance of this algorithm in predicting mJSW against human experts. Several impediments in predicting the mJSW are also discussed, such as the lack of a ground truth when measuring the mJSW and lack of a radiological definition of hip osteoarthritis. Nevertheless, the study revealed artificial intelligence methods to be quite precise in measuring the mJSW. In general, more research has been done in segmenting the knee joint space rather than the hip joint space [4], with high accuracies in predicting the bone outlines [3].

Given the benefits of an automatic tool for hip JSW estimation, the research question being tackled in this publication is the following: ***How accurate is a ResUNet based deep learning approach for predicting the minimum joint space width along the weight-bearing part of the hip joint in a 2D image, in comparison to ground-truth data generated by the BoneFinder algorithm?*** More specifically, the aim of this research is to develop a pipeline for training a ResUNet model and to analyse the performance of this deep learning model in estimating the mJSW. One important feature of this research is the usage of automatically generated labels, as opposed to manually annotated data which can be scarce.

In addition to this, the study will discuss key decisions in terms of how to preprocess and label the data and what configuration of hyperparameters (loss function, dataset sizes, output activation functions etc.) for the U-Net gives the best results in terms of estimating the segmentation masks and the JSW.

2 Methodology

The experiments for calculating the JSW presented in this research employ a deep learning approach, wherein a deep network for image segmentation is trained on automatically generated labels to highlight the hip joint components in X-ray images. Then, an additional algorithm identifies the contours of the segmented joint bones and computes the JSW. The most important steps of our approach are detailed below: label generation, deep network architecture and JSW calculation.

2.1 The BoneFinder Algorithm for Label Generation

Annotating X-ray images for segmentation tasks is often a resource-consuming labour, as it involves human experts to interpret the radiography and to manually highlight the different objects in the image to be segmented.

To this end, the segmentation masks used as labels were generated with the BoneFinder algorithm, a reliable automatic method for bone segmentation [12][11]. Thus, a set of points outlining the pelvic bones for each X-ray image was generated using BoneFinder. From this set of points, only a subset is selected, corresponding to set of the bones to be segmented. Further, the coordinates in this subset are used to draw a polygon representing each bone in the image.

As opposed to an approach which would estimate only the mJSW, the method presented in this research not only estimates the mJSW, but it can also display the region of the joint space where the mJSW was identified using the predicted segmentation masks.

2.2 Network Architecture

The Residual U-Net (ResUNet) architecture used in this research is illustrated in figure 1. The backbone modules of ResUNet are a variation of the U-Net architecture [14] and modified residual blocks of convolutional layers.

U-Net is based on the Fully Convolutional Network (FCN) architecture, following the encoder-decoder design. It is widely employed in medical imaging applications, thanks to its effectiveness in identifying object boundaries and relatively fast training time, being able to reach high accuracy with limited training data when applying augmentation strategies [14].

In ResUNet, modified residual blocks replace the convolution blocks in the U-Net. While convolution blocks learn some output $\mathcal{H}(x)$ given some input x, residual blocks learn the residual $\mathcal{F}(x)$, where $\mathcal{F}(x) = \mathcal{H}(x) - x$. This is done by including a *shortcut connection* which adds the input x to the residual $\mathcal{F}$ produced by the residual block (i.e., $\mathcal{H}(x) = x + \mathcal{F}(x)$). Very deep networks generally suffer from the *"degradation problem"*. Residual learning address this issue, maintaining an increasing trend in accuracy when having greater number of network layers [7].

Similarly to U-Net, the ResUNet architecture, as shown in figure 1, is composed of a contracting path (or encoding path; left side) and an expanding path (or decoding path; right side). Each path consists of 4 residual blocks, with skip connections between homologous encoding and decoding blocks, while a bridge block links these two segments. The network receives a 2D image of size $1 \times 512 \times 512$ as input, with the final output being a multi-class, one-hot encoded segmentation mask of size $N \times 512 \times 512$, where N is the number of classes.

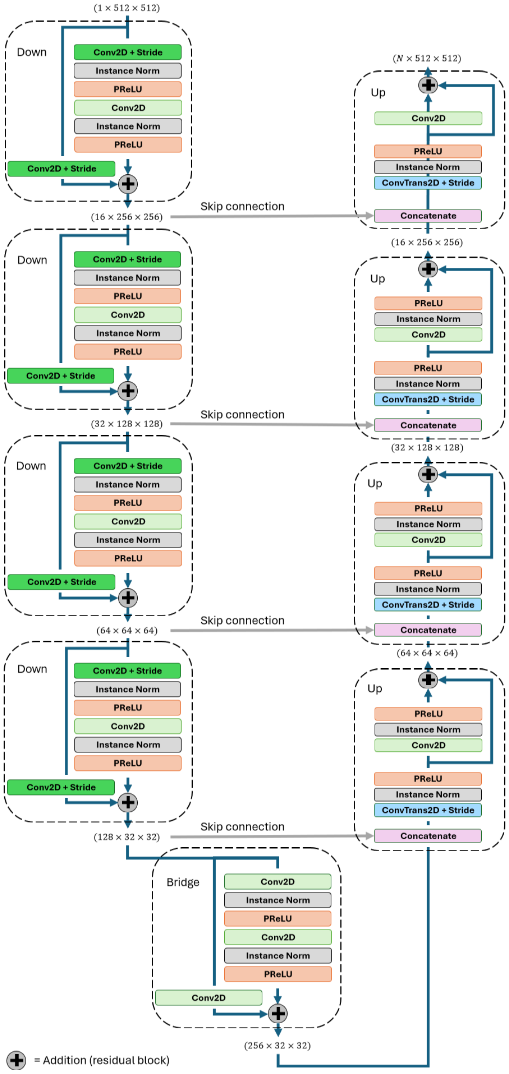


Figure 1: ResUNet architecture. The number of feature channels, height and width of inputs and outputs are displayed before and after each block. Concatenation layers are used to combine the low-level features in the decoding path with high-level features received via skip connections from corresponding blocks in the encoding path.

After each encoding unit, the number of feature channels is doubled, while each decoding unit halves this number. Each block generally consists of two successive applications of the following sequence of layers: convolution, instance normalization and parameterized rectified linear unit (PReLU). Furthermore, convolutions and transposed convolutions with stride of 2 at the beginning of each block are used for downsampling and upsampling, respectively, instead of max-pooling operations. Similarly, the modified residual blocks in the encoding path use convolution layers with stride of 2 as projections in their *shortcut connections*, which downsample the input to match the dimension of the residual. As opposed to U-Net, only padded convolutions with kernel size of 3 × 3 are used, to ensure that the size of the output after each layer is an even multiple of the initial input.

The PReLU and instance normalization layers were omitted in the last upsampling unit. Instead, the experiments in this research used either the sigmoid or the softmax functions as output layer activation. If using sigmoid function, a thresholding transform is applied on all pixels after the sigmoid activation, to obtain the final one-hot encoded mask. Using the softmax function requires applying the argmax and one-hot encoding functions after the softmax to produce the final mask. To avoid backpropagating the discrete pixel labels, thresholding, argmax and one-hot encoding are not applied during training steps.

Lastly, the default loss function is the Dice loss. This research also experiments with the Cross-Entropy and Dice-Cross-Entropy losses, the latter being a linear combination of Dice and Cross-Entropy (i.e., $DiceCrossEntropy = Dice + CrossEntropy$). Tthe Dice metric is used to evaluate the the ResUNet performance.

2.3 JSW Calculation

As mentioned in the Introduction section 1, the currently most common score used in clinics for grading HOA is the mJSW, denoting the distance at the narrowest section of the hip joint. This research is centered on estimating the mJSW based on segmentation masks of the joint space predicted by a deep learning model.

To compute the mJSW, hip articulation bones are first segmented using the ResUNet. Then, using the predicted segmentation mask, the joint space pixels neighbouring the hip articulation bones (acetabular roof and femoral head; see figure 4) are identified as the upper and lower borders of the joint space. Finally, the mJSW is calculated as the minimum point-to-point distance between these borders. It is assumed that the pelvic radiograph is displayed vertically (i.e., from a standing position of the patient).

3 Experimental Setup

The aim of the experiments in this research was to develop an approach to calculate the mJSW, based on image segmentations obtained using the ResUNet model. To this end, five components were created: **data preprocessor**, **data splitter** (performs train-validation-test split), **model trainer**, **model evaluator**, **JSW calculator**, as illustrated in figure Figure 2. The main advantage of this approach is that the enumerated components can be executed sequentially without having to run all previous steps when re-executing a certain component. For instance, the training step can be executed multiple times, without having to preprocess the data with every training session.

3.1 Data Acquisition

The data used in this research are represented by 2D X-ray images stored using the DICOM[1] standard. The total number of images available for this research is 14994, collected from the CHECK (3703 images) [15] and OAI (11291 images) datasets [6].

The ground truth data used for creating the segmentation masks as labels for each sample is represented by a fixed number of 160 pixel coordinates (expressed in millimeters) used to highlight the borders of the various hip bones. These points were generated using the BoneFinder algorithm [12] and they were provided together with the X-ray images.

To ensure protection of patient sensitive data, the images and the BoneFinder points are stored on the servers of the DelftBlue supercomputer [1] and the experiments were executed on this supercomputer.

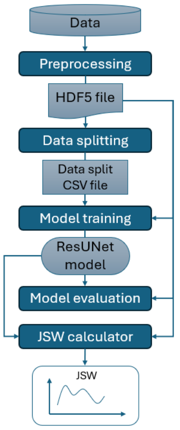


Figure 2: Modules of the experiment pipeline.

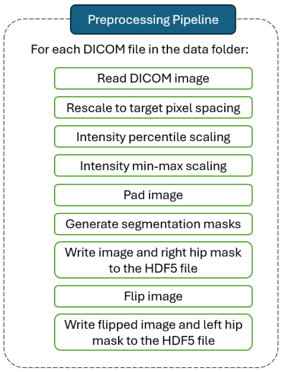

Preprocessing Pipeline
For each DICOM file in the data folder:
Read DICOM image
Rescale to target pixel spacing
Intensity percentile scaling
Intensity min-max scaling
Pad image
Generate segmentation masks
Write image and right hip mask to the HDF5 file
Flip image
Write flipped image and left hip mask to the HDF5 file

Figure 3: Data preprocessing pipeline.

3.2 Data Preprocessing

The preprocessing phase is concerned with processing the X-ray images and creating the segmentation masks based on the BoneFinder

[1] https://www.dicomstandard.org/

points. Figure 3 highlights the main steps for preprocessing.

3.2.1 Image Processing

After reading the image DICOM files from the data directory, several image processing steps are applied on each X-ray. Firstly, image resolution is rescaled to an isotropic target pixel spacing of 0.9 $mm/pixel$, reducing the size of each image and preserving the outlines of the X-ray objects. The target resolution value of 0.9 $mm/pixel$ was chosen to later minimize the amount of padding needed to reach the target shape. Secondly, pixel intensities are scaled by applying the Percentiles and Min-Max normalization transforms, in this order. Percentiles normalization removes outlying pixel values, by bounding the interval of intensities to the 5^{th} and 95^{th} percentile values, as lower and upper bound, respectively. Min-Max normalization standardizes the range of intensities for all images by scaling pixel values to the interval $[0, 1]$. Finally, each image is padded until reaching a target shape of 512×512 pixels.

The preprocessing module also discards images that do not satisfy certain properties. Thus, each X-ray must be stored in a DICOM file as a monochrome image, with colors ranging from black to white (i.e., background is darker, while bone features are brighter). Each image file must specify an isotropic pixel spacing, and those that cannot be rescaled to a target pixel spacing or cannot be padded due to large size are also dropped. Lastly, there must be a corresponding BoneFinder points file for each DICOM file.

From a total of 14994 DICOM images, 14721 were successfully preprocessed. The remaining 273 images were discarded during preprocessing for the following reasons: 134 lacked a corresponding BoneFinder points file, 136 were too large to be padded, 2 DICOM file did not specify the pixel spacing, and 1 image was not stored as a DICOM file. Omitting these samples is considered to have a negligible impact on the training process, given the large amount of available data.

3.2.2 Segmentation Masks Generation

The image segmentation task in this research was concerned with segmenting the following hip structures: the *femoral head*, the *acetabular roof*, and the *joint space*. Training the ResUNet model required labels represented by multi-class, one-hot encoded segmentation masks of the enumerated pelvic structures. In other words, each mask combines multiple submasks (i.e., 2D binary arrays), one for each of the enumerated pelvic components. Submasks classify image pixels as 1 if they are part of their associated hip component and 0 otherwise. The *background* can optionally be included as an additional submask, depending on the model setup to be trained (i.e., softmax activation in the output layer requires the background mask, whereas sigmoid does not). Consequently, the shape of each mask is either 4×512 or $3 \times 512 \times 512$, where the size of the first channel denotes the number of class labels, which in turn depends on whether the background is included. Figure 4 shows an example of a segmentation mask.

The combined masks are generated separately for the right and left hip. The preprocessing step first outputs the image and the right hip mask. Then, for the same sample, it outputs the image again and the left hip mask, but both horizontally flipped. This approach simplifies training by focusing on segmenting the right side of the hip, while providing more training data. In the end, from the total of 14721 successfully processed images, a total of 29442 samples were obtained after preprocessing, by including both the flipped and unflipped images and by having separate masks for the right and left hips.

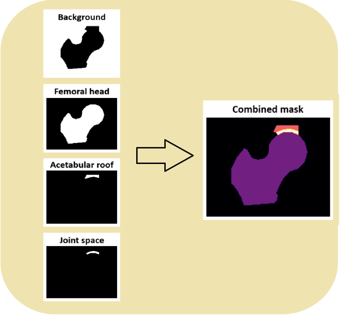


Figure 4: Segmentation mask as a combination of submasks.

3.3 Model Training and Evaluation

Before the training and evaluation steps, the preprocessed data samples are first shuffled, then partitioned into training, validation and testing datasets, generally with ratios of 80%, 10% and 10%, respectively, from the total amount of samples.

The ResUNet model is trained on the training and validation datasets, using a loop which iterates over batches of training data for a specified number of epochs. The training process is combined with a validation step occurring after each epoch. The validation step evaluates the model obtained after the each epoch using the metric function and the validation dataset, saving the model state with the maximum metric score across all epochs. Finally, the trained model is evaluated on the testing dataset, expressing the model performance using the metric function.

A first batch of experiments for training and evaluating ResUNet models with different hyperparameter setups was established, with the aim of determining the best configuration. First, a baseline model format was defined, as described in table 1. Then, a standard testing dataset was created, consisting of 10% of the total amount of samples available (the same test dataset is used to evaluate all model configurations).

Ultimately, in the second batch of experiments, the best model configuration is trained on all available data, then it is used to analyse its behaviour in estimating the mJSW.

Baseline model:	
Network architecture:	ResUNet
Output layer activation:	softmax
Loss function:	Dice
Metric function:	Dice
Optimizer:	Adam
Learning rate:	10^{-3}
Weight decay:	10^{-5}
Training dataset size:	1600
Validation dataset size:	200
Batch size:	20
Number of epochs:	50

Table 1: Baseline model configuration.

3.4 Implementation Details

For the experiments in this research, the pipeline shown in figure 2 was implemented in the Python (version 3.9.8) programming language, due to its ease of development and support for Machine Learning libraries. The *Pydicom* library was used to read the DICOM files, whereas the *HDF5* framework, intended for storage of big data, was used to write the preprocessed images and segmentation masks to a single HDF5 file. The *Scikit-Image* library was used for image processing. *PyTorch* (version $2.3.0 + cu121$) was used for data splitting and for the training and evaluation loops. The *MONAI* framework (version $1.4.dev2418$) [5], specialized in medical image segmentation and compatible with PyTorch, provided the implementations for ResUNet, loss and metric functions. All experiments were executed on the DelftBlue supercomputer [1].

4 Results

4.1 Hyperparameter Tuning

The first experiment trained the baseline model with training and validation datasets of various sizes, revealing that medium size datasets (1600 training and 200 validation samples) can give a testing score similar to large datasets (23554 training and 2944 validation samples) with the same amount of epochs, whereas smaller datasets (160 training and 20 validation samples) required more epochs to converge; see figure 5. Having smaller batches helped the training on smaller datasets to converge faster with the same number of epochs, in a similar fashion to medium size datasets; see figure 6. However, figure 7 reveals that training with medium size training dataset produced a higher test score and faster convergence than a small size one, for the same number of training steps (i.e., number of epochs × number of batches in the dataset).

Secondly, in figure 8, comparing various loss functions indicated that the Dice function performed the best in terms of test score and progression of the validation metric. While Dice-Cross-Entropy showed a similar performance to Dice, Cross-Entropy resulted in a slightly unstable progression of the validation score.

Thirdly, having softmax as the output layer activation function resulted in faster convergence of the validation curve, as opposed to the sigmoid function, even though differences in test metrics were negligible; see figure 9. Moreover, the validation curve for sigmoid activation converges towards a local maximum after 10 epochs, then after the 20th epoch it converges again towards a greater maximum. This might be caused by the small learning rate in the incipient steps of the training and may suggest using a larger learning rate in the first epochs which decays towards a smaller value in later steps. Another sensible reason for choosing softmax over sigmoid is that the latter may assign more than one class label for the same pixel, whereas softmax assigns exactly one label to each pixel. Lastly, the sigmoid function requires a thresholding operation which uses a manually selected threshold, an additional parameter.

In the end, the baseline model trained on all available data was chosen as the best-performing configuration.

4.2 Calculating the min-JSW

The second batch of experiments trained the baseline model on all available data, with ratios of $0.8, 0.1, 0.1$ for the training-validation-testing dataset split, producing a test score of 0.9136. Ultimately, the final model and the test dataset were used to compute and analyse the predicted masks and the min-JSW scores. Table 2 describes the mean standard deviation of the differences between the mJSW scores calculated for real (i.e., generated by BoneFinder) and predicted masks of the test samples, where one sample was dropped due to failed segmentation (see figure 11). Figure 10 shows a typical predicted label (left), but also a failed segmentation (right). Figure 11 illustrates differences in mJSW estimation between the real and predicted labels, but also pixel-to-pixel differences between the two labels.

Mean mJSW	mJSW standard deviation
0.0763	0.0874

Table 2: Mean and standard deviation of min-JSW, calculated for the same test dataset used to evaluate the final model.

5 Responsible Research

As the deep learning tool of this research uses privacy sensitive medical data, the experiments presented in this research were performed on the DelftBlue supercomputer [1], to keep the sensitive images contained only within the remote storage space of DelftBlue. Furthermore, several samples were dropped during preprocessing and evaluation steps, an account of which is given in sections 3 and 4 (i.e., 273 images discarded during preprocessing and 1 sample discarded during mJSW calculation). It was assumed that the pelvic X-rays were displayed vertically, that is, the patient was in a standing position. The code used in the research experiments can be found in the following repository: `https://github.com/iDragos1234/Research-Project`.

6 Discussion and Conclusions

The insights presented in the section 4 firstly compared the baseline model to several modified configurations of itself, proving that the baseline performed the best from the selected setups. The second batch of experiments showed that the model generally performed well in segmenting the X-rays and calculating the mJSW, when compared to BoneFinder labels.

6.1 Segmentation Masks

While the final model produced a large test score and plausible segmentations, there are some important remarks to consider.

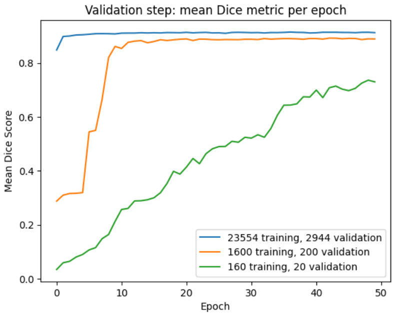
On one hand, the BoneFinder labels do not cover the entire segmented bones and the model may be wrongfully penalized during training due to the arbitrariness with which BoneFinder placed the landmarks outlining the bones. For example, the lower margin of the femur is usually placed well above the region where the femur shaft ends in the images, which is unrelated to the inherent structure shown in X-rays. This problem can be addressed in further work by generating weight maps that avoid loss computation in certain regions of the labels (i.e., loss is non-zero only for pixels with weights of 1 in their binary weight maps).

On the other hand, the central portion of a segmented object contributes with a large portion to the Dice score, whereas the bordering region has less impact. To this end, future research could use the Hausdorff distance as loss and metric functions, which computes the distance between the currently predicted shape and the target shape.

In the end, using the BoneFinder algorithm to automatically generate labels considerably more training data, as opposed to a manual approach in annotating the data,

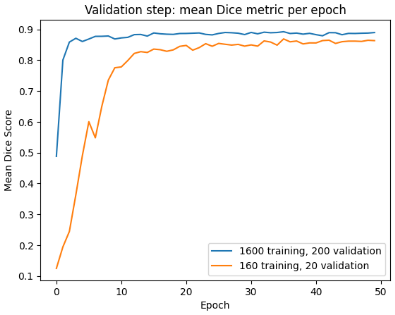
6.2 mJSW Scores

The ResUNet approach presented in this research produced plausible scores for the mJSW, with small differences between scores computed based on the BoneFinder masks and the predicted masks. As opposed to an approach which would estimate only the mJSW, the current method not only estimates the mJSW, but also displays the region of the joint space where the mJSW was identified using the predicted segmentation masks. While the ResUNet is a black-box model in itself, visualizing the region where the narrowest part of the joint space was identified provides a greater degree of explainability. Thus, medical experts can compare the min-JSW computed using the automated tool in this research with their own manual estimations of this score. Additionally, a visualization of the min-JSW region can help with monitoring the HOA progression, as the narrowest region can vary across medical visits for the same patient [13]. Moreover, the human expert may find that the narrowest point of the joint space is located in a different region than the one estimated by the automated tool [13], as the BoneFinder labels do not span the entire joint space. A metric which provides even more detail about the severity of HOA than the min-JSW is the multiple-JSW (or JSW(x)), representing the JSW at an arbitrary position along the articular space while also including the min-JSW. While metric is not in the scope of this research, as min-JSW already provides a clinically relevant score for HOA severity, the segmentation component can be re-used to estimate the JSW(x) in future research.



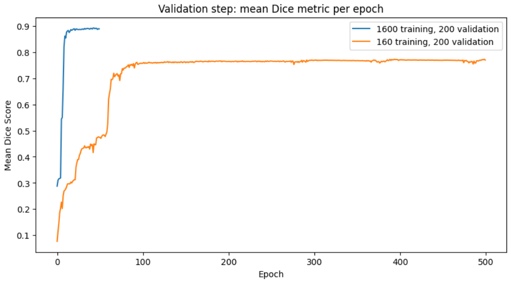
Testing dataset size	Validation dataset size	Test mean Dice metric
23544	2944	0.9136
1600	200	0.8877

Figure 5: Training the baseline model with different sizes for training and validation datasets, with batches of size 20.



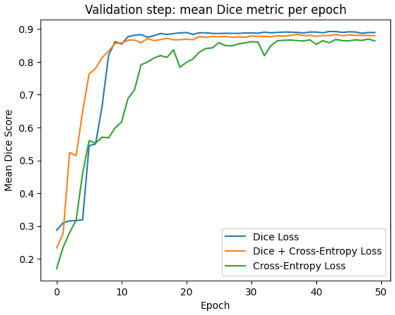
Testing dataset size	Validation dataset size	Test mean Dice metric
1600	200	0.8925
160	20	0.8546

Figure 6: Training the baseline model with different sized datasets and batches of 2.



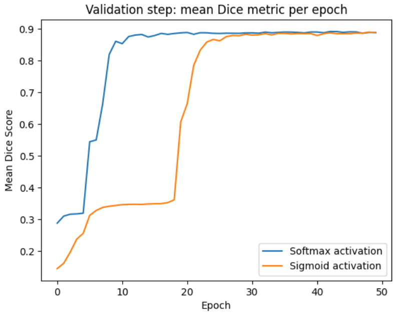
Testing dataset size	Validation dataset size	Test mean Dice metric
1600	200	0.8877
160	200	0.7717

Figure 7: Training the baseline model with same number of steps and equal size datasets for testing and validation, but different sizes for the training one.



Loss function	Test mean Dice metric
Dice	0.8877
Dice + Cross-Entropy	0.8790
Cross-Entropy	0.8672

Figure 8: Training the baseline model with different loss functions. Top: progression of the validation Dice metric per epoch. Bottom: test Dice scores.



Output layer activation	Test mean Dice metric
Softmax	0.8877
Sigmoid	0.8859

Figure 9: Training the baseline model with different activation functions in the output layer. Top: progression of the validation Dice metric per epoch. Bottom: test Dice scores.

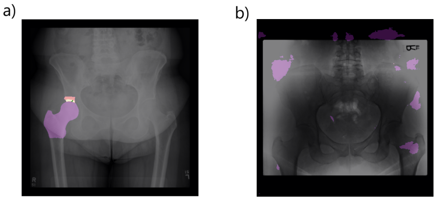


Figure 10: Contrast between a plausible predicted mask (left) and a failed prediction (right). The failed prediction is likely caused by inverted intensities (i.e., background is bright whereas objects are dark).

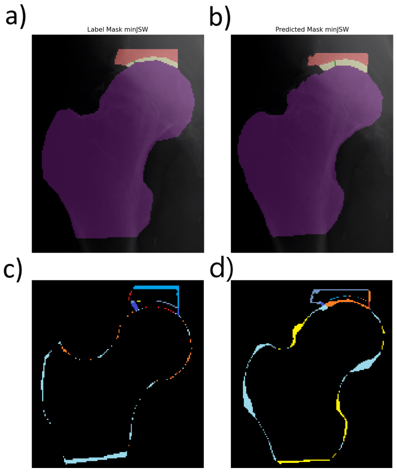


Figure 11: Images **a)** and **b)** show a noticeable difference in mJSW between the BoneFinder mask and the predicted mask: 0.2986 mm and 0.9658 mm, respectively. Images **c)** and **d)** illustrate the pixel-by-pixel difference between the BoneFinder mask (cold colours) and the predicted mask (warm colours).

References

[1] Delft High Performance Computing Centre (DHPC). *DelftBlue Supercomputer (Phase 2)*. https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2. 2024.

[2] Anne Mathilde Andersen **andothers**. ?Minimal Hip Joint Space Width Measured on X-rays by an Artificial Intelligence Algorithm-A Study of Reliability and Agreement? **in***BioMedInformatics*: 3.3 (2023), **pages** 714–723. ISSN: 2673-7426. DOI: 10.3390/biomedinformatics3030046. URL: https://www.mdpi.com/2673-7426/3/3/46.

[3] James Chung-Wai Cheung **andothers**. ?Superiority of Multiple-Joint Space Width over Minimum-Joint Space Width Approach in the Machine Learning for Radiographic Severity and Knee Osteoarthritis Progression? **in***Biology*: 10.11 (2021). ISSN: 2079-7737. DOI: 10.3390/biology10111107. URL: https://www.mdpi.com/2079-7737/10/11/1107.

[4] P.G. Conaghan **andothers**. ?Summary and recommendations of the OARSI FDA Osteoarthritis Assessment of Structural Change Working Group? **in***Osteoarthritis and cartilage / OARS, Osteoarthritis Research Society*: 19 (**march** 2011), **pages** 606–10. DOI: 10.1016/j.joca.2011.02.018.

[5] MONAI Consortium. *MONAI: Medical Open Network for AI*. **version** 1.3.1. **may** 2024. DOI: 10.5281/zenodo.7245505. URL: https://doi.org/10.5281/zenodo.7245505.

[6] F. Eckstein, W. Wirth **and** M. Nevitt. ?Recent advances in osteoarthritis imaging—the Osteoarthritis Initiative? **in***Nat Rev Rheumatol 8*: (2012), **pages** 622–630. DOI: 10.1038/nrrheum.2012.113. URL: https://doi.org/10.1038/nrrheum.2012.113.

[7] Kaiming He **andothers**. ?Deep Residual Learning for Image Recognition? **injune** 2016: **pages** 770–778. DOI: 10.1109/CVPR.2016.90.

[8] J. H. Kellgren **and** J. S. Lawrence. ?Radiological Assessment of Osteo-Arthrosis? **in***Annals of the Rheumatic Diseases*: 16.4 (1957), **pages** 494–502. ISSN: 0003-4967. DOI: 10.1136/ard.16.4.494. eprint: https://ard.bmj.com/content/16/4/494.full.pdf. URL: https://ard.bmj.com/content/16/4/494.

[9] Mark Kohn, Adam Sassoon **and** Navin Fernando. ?Classifications in Brief: Kellgren-Lawrence Classification of Osteoarthritis? **in***Clinical Orthopaedics and Related Research*: 474 (**february** 2016). DOI: 10.1007/s11999-016-4732-4.

[10] Michelle J Lespasio **andothers**. ?Hip Osteoarthritis: A Primer? **in***The Permanente Journal*: 22.1 (2018), **pages** 17–084. DOI: 10.7812/TPP/17-084. eprint: https://www.thepermanentejournal.org/doi/pdf/10.7812/TPP/17-084. URL: https://www.thepermanentejournal.org/doi/abs/10.7812/TPP/17-084.

[11] Claudia Linder. *BoneFinder*. Last accessed 23 June 2024. 2024. URL: https://bone-finder.com/#contact.

[12] C. Lindner **andothers**. ?Fully automatic segmentation of the proximal femur using random forest regression voting? **in***IEEE Trans Med Imaging*: (**august** 2013). DOI: 10.1109/TMI.2013.2258030.

[13] G Neumann **andothers**. ?Health ABC Study. Location specific radiographic joint space width for osteoarthritis progression? **in***Osteoarthritis Cartilage*: (2009).

[14] Olaf Ronneberger, Philipp Fischer **and** Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[15] Janet Wesseling **andothers**. **?**Cohort Profile: Cohort Hip and Cohort Knee (CHECK) study**? in***International Journal of Epidemiology*: 45.1 (**august** 2014), **pages** 36–44. ISSN: 0300-5771. DOI: 10.1093/ije/dyu177. eprint: https://academic.oup.com/ije/article-pdf/45/1/36/6990410/dyu177.pdf. URL: https://doi.org/10.1093/ije/dyu177.