A Computational Method for Histological Bone Research using Convolutional Neural Networks

Daria Yasafova[†], Chris Emezue[†]
[†] Technical University of Munich

Abstract

Histological bone research, which involves measuring the characteristics of a bone (in this paper, we consider exclusively the boundary perimeter (bone perimeter, B.Pm), cross-sectional area (bone area, B.Ar), width (cortical width, Ct.Wi)), has been greatly enhanced by advanced medical imaging like the computed tomography (CT) scans, which has made it possible to get cross-sectional images (or slices) of different areas of a bone. The long bone, which is the major focus of our research, is divided into three parts according to its structure: diaphysis, metaphysis, epiphysis. Efficiently measuring the bone metrics mentioned above relies on being able to differentiate these areas from one another, since the different regions require unique approaches. For example, the diaphyseal region consists completely of the cortical bone and therefore obtaining the bone metrics is not perplexing. On the other hand, when it comes to the area of the metaphysis, and especially epiphysis, it becomes more difficult to distinguish the cortical bone from the spongy bone, and so the task of extracting the bone metrics is hampered. In this paper, we propose a fully automated solution for histological research of diaphysis. For this we designed and trained a convolutional neural network-based model to classify the cross-sectional images. Furthermore, we created computational methods to extract the bone measurements and applied them to bones classified as "diaphysis".

Keywords: bone histomorphometry, automation, CNN, bioimaging, CT scans, computational medicine

1. Introduction

Our proposed computational methods and model will be packaged into a software (and open-sourced), which will perform the monotonous work of checking, analyzing and collecting metrics from a vast number of images. Given the recent advancement in computed tomography (CT) scans and the importance of bone research (e.g., due to age-related changes: mankind continues to increase average life expectancy, which causes progressive bone tissue degradation when reaching a certain age; accidents and other traumatic incidents leading to bone damage), it is also necessary to take care of the means of processing such a huge amount of data.

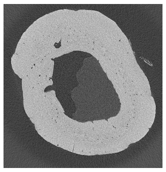

Figure 1. Diaphysis

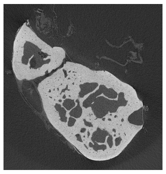

Figure 2. Metaphysis

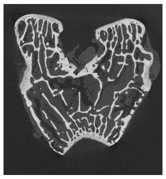

Figure 3. Epiphysis

2. Background

Recent years have witnessed an increased research to optimize automated bone histology [1–3]. The ASBMR Histomorphometry Nomenclature Committee [4] published the standard nomenclature system for bone metrics. [2] did research on bone histology automation for some of the bone metrics. [5] worked on various ways to optimize bone histomorphometry using the ImageJ toolbox [6], a popularly used software for medical image analysis. [3] did an extensive study on the implementation of artificial neural networks to medical image analysis: from discussing neural networks, to explaining the techniques involved in applying them to computer-aided diagnosis, medical image segmentation and edge detection.

This project contributes to the wide research on automated bone histology, by utilizing deep convolutional neural networks [7] and creating unique algorithms to compute some of the bone metrics effectively, especially when many images are involved. Before discussing the project, it is important to give a brief overview of the following bone histology-related terms:

Adult bones can be classified into two different types depending on their porosity: cortical bones and spongy bones [8]. The microstructures of these bones are changed by means of load adaptive remodeling to meet the mechanical requirements. Consequently, osteons in dense cortical bone and trabecula in porous spongy bone are aligned along the direction of local loading. It is believed that the difference in the porosity of the cortical and spongy bone is due to the difference in the local balance between bone formation and resorption. On the one hand, bone formation, i.e. an increase in the volume of mineralized bone, is associated with an improvement in the bearing function of mineralized bone. On the other hand, bone resorption, i.e. increase of bone marrow volume, is connected with improvement of material-transport function of bone marrow.

The long bone can be divided into three parts according to its structure: diaphysis, metaphysis, epiphysis. The epiphysis is located at the ends of the bone, closely followed by the metaphysis, while the diaphysis occupies the middle. *Diaphysis* is the main and longest part of the bone. It consists entirely of the cortical bone, containing the medullary cavity, which stores the red and/or yellow marrow (so it is also known as the bone marrow cavity). The *metaphysis* is a narrow part of the long bone between epiphysis and diaphysis. It has a thinner area of cortical bone, a larger spongy bone and a wider corresponding diaphyseal part. The porosity in the metaphysis region is larger than in the diaphysis region, and the periosteum is more firmly fixed in the metaphysis region as it approaches the epiphysis [9]. The *epiphysis* lies between a growth plate (physical) and a joint at the end of the bone. In mice and humans, most tubular bones develop epiphysis at both ends. Some bones, such as phalanges, metacarpal and metatarsal bones, usually only form this structure at one end [10].

The *cortical bone* surrounds the diaphysis and metaphysis of long bones and is a solid matter with a porosity of approximately 5%. The *spongy bone part*, also sometimes referred to as trabecular, is usually found at the ends of long bones and within the vertebral bodies. Spongy bone (Click to see figure) is highly porous (>70-80%) and consists of more voids than bone (Click to see figure), which resembles a lattice in structure. The large surface area contributes to the fact that the spongy bone has a higher metabolic activity than the cortical bone. Although it is believed that the cortical bone carries most of the load in an individual bone, spongy bone contributes and serves to distribute the load to the cortex. The effect is achieved by the sponge network and the individual trabecula. Individual human trabeculae are 100-200 μm thick and have shapes ranging from cylindrical rods to flattened plates. The cortical bone is approximately 80% of the skeletal mass and the spongy bone is approximately 20%.

3. The Project

This project is divided into two parts: in the first part, we train the neural network to recognize and classify bone slices into two categories: diaphysis, and not-diaphysis (metaphysis or epiphysis) areas. In the second part, using the images classified as diaphysis, we propose computational algorithms to estimate the bone metrics.

3.1. Part 1: Implementation of convolutional neural networks - DARY

In this stage, we design and train a model to learn to classify the bone cross-sectional images into "diaphysis", and "not-diaphysis". In the following sections we will describe the data collection, preprocessing, training and results.

Data Collection and Preprocessing

The dataset was collected from O.A. Sachenkov's research group [2, 11]. The images are cross-sectional slices of mouse bones obtained through CT scans. Due to the complexity of finding standard bone characteristics in areas such as metaphysis and epiphysis (although the extraction of these characteristics from the diaphysis area is not a perplexing one), namely, obtaining incorrect data, due to the fact that bones in such slices may look quite differently (View images 1 and 2), the complexity of interpretation of the image itself, i.e. finding the boundary between the spongy and cortical bones, the difficulty of determining where the metaphysis ends and the epiphysis begins, we decided to design and train a model to make this classification.

The original data were stored as ".vol" files, each representing a three-dimensional image of a bone (examples of 3D visualizations can be seen here: 1, 2, 3). Due to the fact that such three-dimensional images are obtained by stacking or superimposing two-dimensional slices on each other, it was possible to "cut" the three-dimensional image into two-dimensional slices for our study.

In order to successfully divide a three-dimensional image into two-dimensional slices, it was important to know the data format (16-bit unsign int) and image dimensions. For example, "718 749 933" meant that from this three-dimensional file we could get 933 two-dimensional slices of 718×749 size. All the two-dimensional, cross-sectional images obtained (a total of 8556) were saved in grayscale format, which were resized to (224, 224), in order to match the dimensions of the pretrained model. Data augmentation (in the form of random horizontal/vertical flips, and rotations) were made.

Training of DARY

DARY — uses the pretrained EfficientNetB0 model[1], based on the EfficientNet [12] architecture, as a feature extractor, with the fully connected layer of the EfficientNetB0 model switched with a new one with two classes for ["Diaphysis","Not Diaphysis"]. We used this pretrained model because it has been shown to give best results on CT image classification [13, 14].Weights (for the final fully connected layer) were initialized with He initialization [15]. The size of each image is (height=224, width=224, number of channels=3). We divided the samples as follows: 5455 samples for training, 2139 for testing, and 962 for validation. We made sure that both classes were equally represented in each of the three categories (for training, validation, and testing) to avoid imbalance. To compile the model, we used the Adam[2] optimizer with a learning rate of 0.001, and an exponential learning rate decay (after 10 epochs). We also experimented with the ResNet50 [16]), but got better test results with DARY. DARY has a total of 5.3M parameters. We trained DARY for 102 epochs (small epoch since we are performing feature extraction with the pretrained model). All training was done with PyTorch and the code is available here.

[1] https://github.com/lukemelas/EfficientNet-PyTorch

[2] https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam

Results of DARY

Although DARY was trained on few samples (and few epochs), it was able to learn the distinctive features of diaphysis, and non-diaphysis in bone CT images. On the testing samples, DARY had an accuracy of 64.64%. The table 1 shows the performance of DARY on images from the test data set (i.e., on which no training was conducted). Furthermore, our independent human evaluations on several samples (checks were carried out for each class many times, but we gave only one example for the sake of brevity) demonstrated, as shown in Table 1, that DARY is capable of detecting abstract distinguishing features of images with and without diaphysis, and can tell whether the slice belongs to diaphysis or not.

Table 1. Predictions from DARY for sample test images

Test Images (Click to view Image)	**True Class**	**Predicted Class**
Image 1	diaphysis	diaphysis
Image 2	not-diaphysis	not-diaphysis
Image 3	not-diaphysis	not-diaphysis

3.2. Part 2: Proposed computational methods for extracting bone metrics

Our goal is to extract the following bone metrics: the boundary perimeter (bone perimeter, B.Pm), cross-sectional area (bone area, B.Ar) and width (cortical width, Ct.Wi) in the images classified as "diaphysis" by DARY. Originally the bone images (slices) (see figure) are grayscale images, and their pixel values are in the range from 0 to 65535. In order to work with an image, we first convert each value to either 0 or 1 and thus get a black-and-white image (see figure). This process is called threshold binarization. For each image we find a value (threshold) $m + 246\alpha$, where m — is the average pixel value for the current image. We can vary the α parameter so that we get different black and white images. This method is global thresholding.

If we set α big enough, we get rid of white dots outside the bone which are of no interest to us, but simultaneously we also get more black dots on our bone. To get rid of the resulting black holes, we add layers of white dots to all the existing white point boundaries, thus filling the holes. Although this method successfully gets rid of holes, it also adds several layers of white dots to the bone, making the bone thicker, the perimeter of the outer boundary bigger and the inner smaller, i.e. distorting the original data. Therefore, several stages of adding white point layers are followed by several stages of removing white point layers to return the bone to its original shape (compare before and after).

Comparison of OpenCV: OpenCV[3] doesn't work directly with .vol files. We worked on raw (.vol) files and wrote algorithms to work with raw pixels directly. There are many stages in our work (binarization, filling holes, finding boundaries, grouping points) and for each of them we used our algorithms. Applying OpenCV to that particular stage would require converting each 2D slice of an initial 3D raw file to jpg, then filling holes using erode and delude functions, and then converting back to continue working with raw pixels, which is time consuming. Erode and delude functions don't offer a better result than ours. From the results we got, closing (opening) operation is better, as it doesn't distort boundary line, while filling holes. But for these functions to give optimal results, the hyperparameters need to be manually tuned, which is contrary to our goal of automating the process.

[3] https://docs.opencv.org/master/d4/da8/group__imgcodecs.html, Accessed: 20.04.2021

To find the cortical bone area, we count the number of white dots in the image, then we convert the obtained number knowing the resolution of one pixel, e.g. 0.006986 mm. × 0.006986 mm to find actual area.

To find the perimeter, we need to first find the bone boundaries. Our bone is a set. In order to find boundary of a set we exclude interior points. Using the definition of an interior point: *A point a is called an interior point in A, if* $\exists \varepsilon > 0 : O_\varepsilon(a) \subset A$, we check all white points in the image. If the current white point (x, y), among the points $(x+1, y), (x-1, y), (x, y-1), (x, y+1), (x-1, y-1), (x+1, y-1), (x-1, y+1), (x+1, y+1)$ surrounding it, has $< \beta$ white dots, then we put it in the list of boundary dots, if not, we make it black. Any number from 1 to 8 can be taken as β: if $\beta = 1$, the whole image will be painted black; if $\beta = 8$, the boundaries will be very dense and delineated, but the number of points will also be large, which will increase the computation time of the next stages, which will be discussed later. If $\beta = 6$, then the boundary points will be very close to each other, albeit much smaller in number. The resulting set of boundary points (see figure) should be divided into groups of boundary points: the outer boundary of the bone, the inner boundary, etc. For this purpose, we created the following algorithm which utilizes the distance between the points.

Algorithm:

Suppose we have a set of boundary points W, consisting of points w_0, w_1, w_2, ..., w_n. We take the first point, w_0, and mark it with a_0. Then we add a_0 to the new group of boundary points A, remove it from W, and count the distance from a_0 to each of the remaining points $\{w_1, w_2, ...w_n\}$. The point, w_s, that gives the smallest distance is marked with a_1, added to A and removed from W. Now we find the distances from a_1 to $\{w_1, w_2, ..., w_{s-1}, w_{s+1}, ..., w_n\}$. We take the point corresponding to the smallest distance, w_l and similarly mark it with a_2, add it to A, and remove it from W, etc. We continue this iterative process until the smallest distance is more than γ (an asigned value to prevent moving beyond the boundary being considered) or we come across the point a_0, which means that we are back to the beginning of the boundary. In both cases we should create a new group of points, take the first of the remaining W points and perform the same iteration until we reach the same stopping criteria. We continue this process until we have exhausted all the W boundary points.

The total number of boundary points n affects the speed of the algorithm as distances should be calculated $\sum_{i=1}^{n} i$ times. So it is preferable to set β as low as possible. Empirically, we have found that $\beta = 6$ is optimal for subsequent grouping of points, since the points within one boundary are close enough to each other. This is not the case for $\beta = 5$, which may lead to incorrect groupings; points of the same boundary can be mis-classified as belonging to different boundaries, leading to a "torn" boundary line, which occurs when one boundary is broken into several disconnected curves. As a result of running the algorithm, all boundary points from W will be divided into groups. Then, based on the fact that in the diaphyseal zone the only possible situation is when there is one inner and one outer boundary, the boundaries we are interested in are therefore the two biggest clusters of the grouped points, and the greater boundary of the two is the external, since it has a larger diameter.

4. Conclusion

In this paper we proposed a solution to the task of extracting bone measurements from cross-sectional CT scans. First we introduced our model, DARY, trained to classify the cross-sectional images into two categories of "diaphysis", and "not-diaphysis". We further developed algorithms and methods to calculate some bone measurements (the boundary perimeter (bone perimeter, B.Pm), cross-sectional area (bone area, B.Ar) and width (cortical width, Ct.Wi)) of the processed images marked as *diaphysis* by DARY. Also, a method for filling small bone inclusions (black areas), through several stages of layering of white dots, followed by several stages of layer removal, was proposed (see Section 3.2). This makes it possible to get rid of unwanted areas virtually without changing the shape and measurements of the bone (thickness, perimeter of boundaries). This method facilitates subsequent processing of the image at the next stage of grouping boundary points, especially when the inclusion is located close to the outer or inner boundary. To promote reproducibility, our model and codes will be open-sourced on our Github page (link will be provided after the review).

References

[1] D. Malhan, M. Muelke, S. Rosch, A. B. Schaefer, F. Merboth, D. Weisweiler, C. Heiss, I. Arganda-Carreras, and T. El Khassawna. "An Optimized Approach to Perform Bone Histomorphometry". In: *Frontiers in Endocrinology* 9 (2018), p. 666. ISSN: 1664-2392. DOI: 10.3389/fendo.2018.00666. URL: https://www.frontiersin.org/article/10.3389/fendo.2018.00666.

[2] V. V. Yaikova, O. V. Gerasimov, A. O. Fedyanin, M. A. Zaytsev, M. E. Baltin, T. V. Baltina, and O. A. Sachenkov. "Automation of Bone Tissue Histology". In: *Frontiers in Physics* 7 (2019), p. 91. ISSN: 2296-424X. DOI: 10.3389/fphy.2019.00091. URL: https://www.frontiersin.org/article/10.3389/fphy.2019.00091.

[3] J. Jiang, P. Trundle, and J. Ren. "Medical image analysis with artificial neural networks". en. In: Computerized Medical Imaging and Graphics 34 (2010), 617 – 631,

[4] D. Dempster and J. Compston. "Symbols, and Units for Bone Histomorphometry: A 2012 Update of the Report of the ASBMR Histomorphometry Nomenclature Committee". en. In: Journal of Bone and Mineral Research (2013).

[5] D. Malhan, M. Muelke, S. Rosch, A. B. Schaefer, F. Merboth, D. Weisweiler, C. Heiss, I. Arganda-Carreras, and T. El Khassawna. "An Optimized Approach to Perform Bone Histomorphometry". In: *Frontiers in Endocrinology* 9 (2018), p. 666. ISSN: 1664-2392. DOI: 10.3389/fendo.2018.00666. URL: https://www.frontiersin.org/article/10.3389/fendo.2018.00666.

[6] J. M. M. Perez and J. Pascau. *Image Processing with ImageJ.* Packt Publishing, 2013. ISBN: 1783283955.

[7] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio. *Object Recognition with Gradient-Based Learning.* nl. Berlin, Heidelberg: Springer-Verlang, 1999.

[8] Y. Kameo, N. Sakano, and T. Adachi. "Theoretical concept of cortical to cancellous bone transformation". en. In: Bone Reports (2020).

[9] S. Frick. "Skeletal Growth, Development, and Healing as Related to Pediatric Trauma". en. In: Greens Skeletal Trauma in Children (2015).

[10] C. Jerome and B. Hoch. pt. In: *Skeletal System.* Comparative Anatomy and Histology (2012).

[11] D. Baidamshina, S. E., S. O., B. T., Z. R., and K. A. *Automatic processing and quantification of histological images.* en. FEBS Open Bio, 2019, – 9, – 312–313. URL: https://repository.kpfu.ru/?p_id=213654.

[12] M. Tan and Q. V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *CoRR* abs/1905.11946 (2019). arXiv: 1905.11946. URL: http://arxiv.org/abs/1905.11946.

[13] P. Gifani, A. Shalbaf, and M. Vafaeezadeh. "Automated detection of COVID-19 using ensemble of transfer learning with deep convolutional neural network based on CT scans". In: *International Journal of Computer Assisted Radiology and Surgery* 16.1 (2021), pp. 115–123. ISSN: 1861-6429. DOI: 10.1007/s11548-020-02286-w. URL: https://doi.org/10.1007/s11548-020-02286-w.

[14] S. Ahuja, B. K. Panigrahi, N. Dey, V. Rajinikanth, and T. K. Gandhi. "Deep transfer learning-based automated detection of COVID-19 from lung CT scan slices". In: *Applied Intelligence* 51.1 (2021), pp. 571–585. ISSN: 1573-7497. DOI: 10.1007/s10489-020-01826-w. URL: https://doi.org/10.1007/s10489-020-01826-w.

[15] K. He, X. Zhang, S. Ren, and J. Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV).* 2015.

[16] K. He, X. Zhang, S. Ren, and J. Sun. *Deep Residual Learning for Image Recognition.* en. 2015.