ISSN 2278-3091

Volume 8, No.1.6, 2019

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse1381.62019.pdf

https://doi.org/10.30534/ijatcse/2019/1381.62019

Acceleration of Knee MRI Cancellous bone Classification on Google Colaboratory using Convolutional Neural Network

Mazhar Javed Awan[1], Mohd Shafry Mohd Rahim[2], Naomie Salim[3], Ajune Wanis Ismail[4], Hassan Shabbir[5]

[1,2,3,4]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia (UTM), Johor, Malaysia

[1,5]Department of Software Engineering, University of Management and Technology, Pakistan

ABSTRACT

The field of Biomechanical engineering and Orth ology is related to knee bone tissue in which cancellous bone lies. The cancellous bone also called spongy bones has greater surface area for this it causes Osteoporosis (OP), Osteoarthritis (OA), and knee cartilage and knee replacement. The knee bone images are measured mostly by Magnetic Resonance Imaging (MRI).In this paper we presented deep learning model on cancellous bones (tiff type) MRI through Convolutional Neural Network (CNN) to predict the image classification which achieved 99.39 % accuracy. The sample size of images are 185 cancellous MRI and 185 tiff images. Further we trained our model on cloud service that is Google Colabaratory (Colab) which is Graphical Processing Unit (GPU). The accuracy of this model is same but the execution time per min decreases on GPU environment. We increased the no of epochs 20 then 50 its execution time is 10 times less than CPU. The execution time on GPU google Colab is 2.23 (mins) and on CPU its 24.23(mins).

Key words : Image Classification, Deep learning, MAgnetic REsonance Imaging, Convolutional Neural Network, Google CoLab, Cancellous bone

1. INTRODUCTION

The bone consists of two forms one is organ of body also called femur or tibia and other one which makes most of mess of the bone is bone tissue or osseous. Further the bone tissue is divided (figure 01) into spongy bone or trabecular bone or cancellous bone and other is compact bone or cortical bone [1]. The cancellous bone has higher surface area to mass ration due to less dense. It makes softer, weaker and more flexible characteristics. These bones are typically found on end of long bones, top of femur, tibia and within the interior of vertebrae [2].Moreover these are highly vascular and frequently contains red bone marrow where the production of blood cells, occurs and functional unit of cancellous bone is a trabecula [3].There is no doubt about this now clinical and medical significance of cancellous bone increased in osteoporosis (OP) , osteoarthritis (OA) and knee bone fracture because it more affected than cortical bone [4].

The Deep learning is now using in many fields like natural language processing, image analysis, big data and expert systems. Now a day many of Doctors and physician are very much interested to predict, estimate various medical diseases with the help of deep learning models [5].

The bone images of knee and trabecular bones are mostly in the form of Computer Tomography (CT), X-Rays and Magnetic Resonance Image (MRI) which helps doctors, sportsmen to diagnosis and identifying features through Artificial Intelligence, Machine learning and specially in Deep learning [6].

There are so many applications in terms of MRI image segmentation, image registration, image detection and image classification [7]. Magnetic resonance images have a great role in diagnosis, treatment and improvement of joint disease such as arthritis. Arthritis consists of progressive degeneration (figure 2) of both the bone and soft tissues [8].

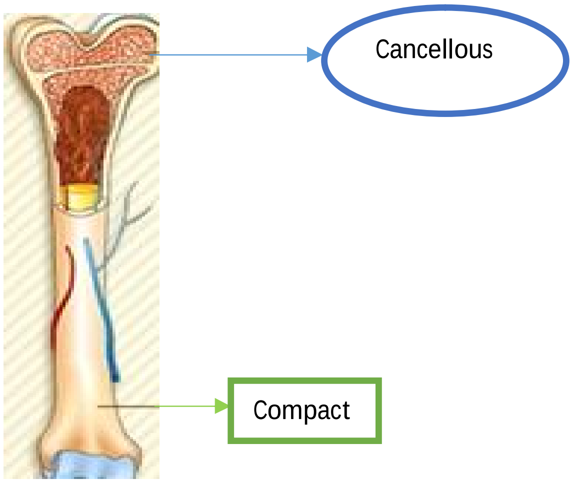


Figure 1: Types of Bone Tissues

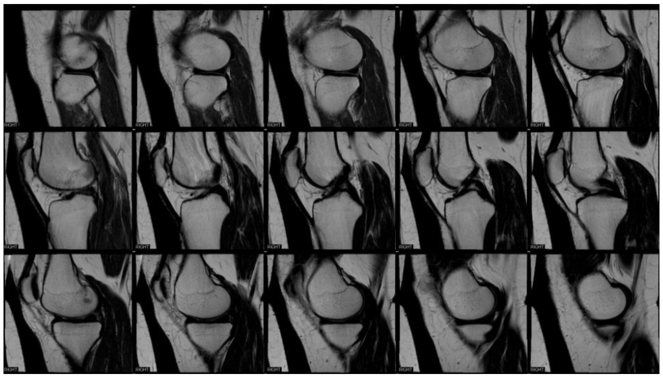


Figure. 2: MRI Knee bone

Deep learning has many models like Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short- term memory Networks (LSTM), AutoEncoders and Generative Adversarial Networks (GAN) [9]. The classical method among these is CNN which is better for image classification, image registration and image segmentation regarding knee bones [10-11]. No doubt about this that deep learning automatically do features engineering but the computational time is major problem in case of radiology images specially in medical and health science [12]. So there is a need to evaluate these images with better accuracy as well as less time like in GPU instead of CPU. But GPU has an expansive solution for researchers and students, for this Google Colaboratory in cloud is better solution in term of free GPUs TPUs, stored data in google drive and write code in Jupyter notebook in python using tensor flow and keras and OpenCV packages for deep learning application [13]. The remaining parts of paper is divided into as follows. Section II is related to literature review, Section III Material and Methodology are many subsections Data, Image Uploading, Building Model, Image generation, Model Architecture, Summary of model and Network of Model. Then Section IV belongs to results and evaluation and last section V is conclusion.

Currently there are many combined literatures work machine learning and deep learning from medical images which taken from Medical/health sciences, orthopedics and biomechanical engineering. There are many surveys regarding machine learning where researchers recognizing patterns in medical images [14-15]. Now the deep learning is much growing field instead of machine learning, the surveys[16-17] on medical image analysis related to image classification , image segmentation, image registration, image augmentation and image diagnostic here researchers used CNN model. The survey related to MRI on deep learning [18] which focused on almost all models of deep learning related to image processing, image retrieval, image segmentation and image prediction.

However, in these above surveys there is not too much work available in the field of biomechanics in terms of machine and deep learning specific to Knee MRI.A very few work is available to classify the Knee MRI for diagnosis of osteoporosis, osteoarthritis and knee joint replacement. The worked on osteoarthritis knee bone segmentation with deep learning using CNN models but their accuracy are not better and it takes lot of computation time in CPU [19-20]. The comprehensive survey on Graphical Processing Units (GPU) [21] taken radiology images of computed tomography (CT), MRI and ultrasound of medical image segmentation have done parallel processing ways with less computation time. The problem of GPU is affordability for a researcher and students in academia. The state of art solution of GPU is best when we can train our model freely for this the study of Google Colaboratory (Colab) is a cloud service, the study here [22] compare and evaluate in terms of hardware resources with CPU, performance of deep learning and computer vision applications of object detection/classification and object segmentation. Moreover here it faster to train the CNN model on CoLab which has 20 physical cores of Linux server [23]. The only limitation of this is Virtual machine which disable after 12 hours.

2. RESEARCH METHOD

2.1 Data Preparation

In this study we taken the cancellous bone MRI images from Sport Innovation Technology Centre (sitc) Faculty of Bioscience and Medical Engineering UTM Johor Bahru. There are 185 MRI files, image type is tiff, number of rows are 1024 and number of columns are 1280 of each image. The image sample of cancellous bone from MRI is in figure 3(a) and other image is same tiff but not MR is in figure 3(b).

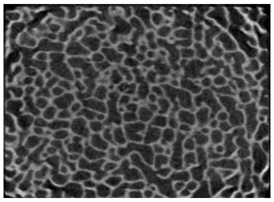


Figure. 3(a): Sample MRI inner

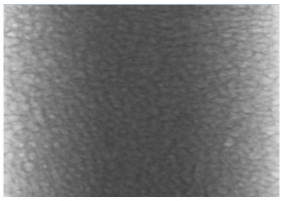


Figure. 3(b): Sample of direct tiff

Coming towards the Implementation, Databricks and Apache spark were the initial setups that has been implemented. Python (3.6) is used here with Karas library to make predictions. After setting up the whole environment and implementing the algorithm, next step is to process data. We used TensorFlow to transform images on numeric features. TensorFlow can be described as a library. It is basically an open source programming library, which can be used for computation purposes [24].

2.2 Image Uploading

The first step to applying deep learning on images is the ability to load the images. To use cloud platform such as Google colab, which supports Google drive we uploaded your dataset on Google drive.

2.3 Image Preprocessing

Firstly, we resized our images to 128 x 128 size and defined two classes labeled as *can* and *tiff* and covered all images in to grey scale image also we split our training dataset by the test size of 0.3 out of 1 to deal with two dimensional images in our dataset we used Convo2D layer by adding 32 number of output filters and 3, 3 width and height of two dimensional convolutional window and input height and width is set to 128 x 128 and channel is set as 1 to deal with grey scale images.

2.4 Building Model

Convolutional Neural Networks are neural designs that are designed specially to handle image data. it was known that high performance is a trade mark of CNNs. This algorithm is very efficient at many tasks. Considering some examples, handwritten digit classification and face detection are a part of them. Convolutional Neural Networks are composed of five basic blocks.

1) Input layer
2) Convolutional layer
3) ReLU layer
4) Max-Pooling layer
5) Fully Connected layer

Input layer: Image is processed as an input and spatial structure is generated and then stored.

Convolution layer: Feature maps are extracted from these inputs. Specific patterns are pointed out from all these maps.

ReLU layer: It basically sets a Zero (0) value for the negative pixels and non-linarites in the network are introduced.

Max-pooling layer: This step is basically to fit the feature maps. It down samples them to do it.

Fully-Connected layer: This layer is to learn the non-linear combinations. These combinations belong to feature maps. The task here is to classify these feature maps.

Building a MRI bones scan classifier using Tensorfow and Keras: Data was organized and stored in Google drive for the ease to process it properly with the help of keras.

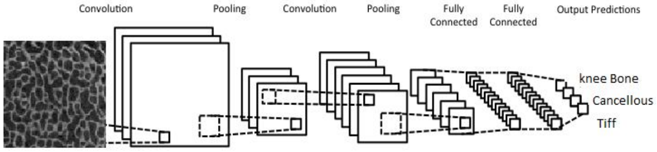


Figure 4: Full CNN Architecture

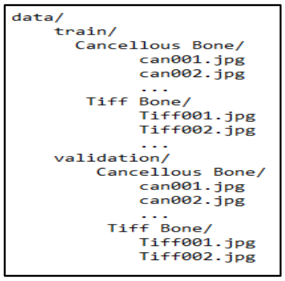
For this purpose, a main folder was created and inside it, there were two other folders called:

1. Train it
2. Validations

Each of them will have 2 folders:

1. Cancellous bone
2. Tiff Bone

The structure is shown in figure 5 as following:



```
data/
    train/
      Cancellous Bone/
            can001.jpg
            can002.jpg
            ...
       Tiff Bone/
            Tiff001.jpg
            Tiff002.jpg
            ...
    validation/
         Cancellous Bone/
            can001.jpg
            can002.jpg
            ...
          Tiff Bone/
            Tiff001.jpg
            Tiff002.jpg
```

Figure 5: Structure of Bone

The structure here is basically to allow this model to understand that from which directory it has to dig the images along with their labels for training purpose or for validations.

To make a file tree we used two parameters

1: Total # of images N

2: ratio of validation set R

2.5 Image generators and data augmentation

During the process of training our model, we used ImageDataGenerator class. This basically provided us the ability to stream the images in a very helpful manner. The images were basically streamed on the bases of the folders names training and validations. There were batches of images and the flow implemented was to process every single batch making forward and backward propagations. There was also a parameter whose job was to update itself. After this process was done, it was the turn of next batch to come and do the same thing with it.

The ImageDataGenerator object used here is to introduce random modifications on each batch. This is basically called as data augmentation. This is done for generalization purposes and helps us to prevent from processing the duplicate data. The data is generalized better and processed better. Two objects of ImageDataGenerator were needed here, which were named as training_dataGenerator and validations_dataGenerator. Both of them were used for the rescaling proposes but train_generator was proposed to done a lot of more modifications. From objects we created two file generators:

- training_dataGenerator
- validation_dataGenerator

Each of them is used to generate batches of images from their respective directory with real-time data augmentation. The data was to process in a loop which goes over indefinitely.

2.6 Model architecture

CNN here was used with total five working layers. Three of them were convolution layers and two were fully connected layers. These three convolutional layers will work with their filters respectively 32, 32 and 64 3x3. To avoid over fitting we used dropout on the other two layers which are fully connected ones.

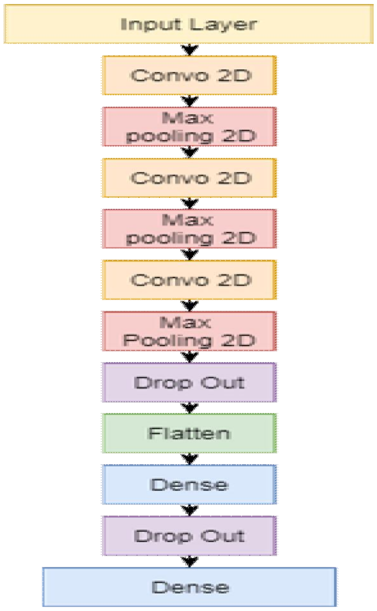


Figure 6: Network Architecture

Relu is used Activation function for this model with pool size of 2, 2 and value for dropout date is set as 0.5 and we are adding flatten layer which is taking output of dropout layer as input and flattening it into one dimensional tensor which is then added in dense layer which has two nodes and softmax activation function. To compile model this study used adadelta optimizer. This model is executed for 7, 20 and 50 epochs and saved into two formats i.e Jason and .h5. One optimizer named stochastic gradient was needed here. This optimizer has a very efficient learning rate that is 0.01 with a momentum of 0.9. As we are going for binary classifications, will be using here cross entropy loss function.

2.7 Training the model

Before training the model, we have defined two callback functions that will be called while training. Then train model for the classification of MRI images of bones by using convolutional neural network this study used sequential model of Keras Library and we also imported Dense, Dropout, Activation, Flatten, Conv2D, MaxPooling2D layers for keras library.

1. Whenever the loss function stops improving on the validation data, early stop the training
2. For storing the validation loss and accuracy of each epoch: this allowed plotting the training error

We used the fit_generator method that takes a generator as input.

3. RESULTS AND ANALYSIS

This study reached 99% accuracy (on the validation data) in 07 epochs (training/validation error and accuracy displayed below) both in CPU based as well as GPU Cloud Google Colab. Here we got results based on train where support is 296 belongs to class 0 and 55 belongs to class 1 in table 01 shown below.

Table 1: Result on training set

	Precision	Recall	F1-Score	Support
Class 0	1	1	1	296
Class 1	1	1	1	55
Micro avg	1	1	1	351
Macro avg	1	1	1	351
Weighted avg	1	1	1	351

The Table 2 shows the result based on cancellous bone Cann class and other one tiff bone tiff class which is validation on our datasets.

Table 2: Result on validation set

	Precision	Recall	F1-Score	Support
Cann	1	1	1	708
tiff	1	0.99	0.99	190
Micro avg	1	1	1	898
Macro avg	1	0.99	1	898
Weighted avg	1	1	1	898

Our findings shows on table 04 that as the number of epoch in increasing execution time when we are training our model on a local computer CPU. The specifications of our local Central Processing Unit (CPU) and Graphical Processing Unit (GPU) is shown in table 03.

Table 3: Comparison of CPU and Google GPU

CPU VS GPU (Google CoLab)	**Specifications**
Our Local CPU System	**Hardware:** Processor Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, 2592 Mhz, 4 Core(s), 8 Logical Processor(s) Total Physical Memory 15.9 GB Operating System: Microsoft Windows 10 Professional **Software:** Distribution: Anaconda Programming Language: Python3
Google CoLab	**Hardware:** GPU: 1xTesla K80 , having 2496 CUDA cores, compute 3.7, 12GB(11.439GB Usable) GDDR5 VRAM CPU: 1xsingle core hyper threaded i.e(1 core, 2 threads) Xeon Processors @2.3Ghz (No Turbo Boost) , 45MB Cache RAM: ~12.6 GB Available Disk: ~320 GB Available **Software:** Distribution: Anaconda Programming Language: Python3

To overcome this issue we can use cloud service for this study we used Google CoLab Cloud where table 04 initial execution time period is much more then later ones. This is because dataset is being loaded for the first time after that re execution time on different number of Epochs is much lower can compared to first one as well as execution time of local environment. If we ignore dataset loading time then 07 Epochs are executed in 20 Seconds and other are executed on same time as before. The performance of local environment CPU and GPU is shown in bar graph where when we increase no of epochs then GPU based system takes only 2,23 min as compared to 24.13 mins in CPU as shown in figure 07.

Table 4: Execution Time (mins) against Epochs

No of Epochs	Local Environment (mins)	Google Colab (mins)
7	3.97	11.6
20	10.37	1.26
50	24.13	2.23

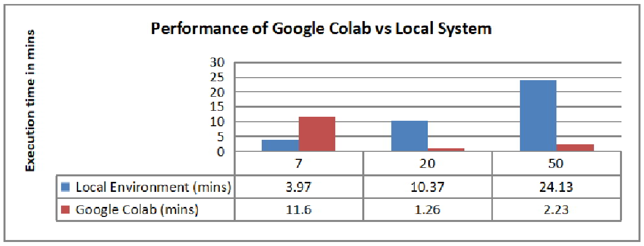


Figure 7: Performance bar graph CPU vs Google Colab (GPU)

4. CONCLUSION

This study has helped to classify MRI scans of human's bones belonging different classes by using Convolutional neural network (CNN) algorithm. This study also proposed that due to increase in the volume of health care data it is very good practice to train our model at large scale and fast processing by using cloud platform Google colab .The study achieved 99.39 percent accuracy on over data set. Till now deep learning applications are providing very positive results but due to the sensitivity of health care data we should try look for more advanced methods of deep learning that can deal with more complex features on very minor details. We can go for the mixing of the different types of networks together into hybrid models. Also we can get benefit of in memory processing by using apache spark

ACKNOWLEDGEMENTS

This research is sponsored partially by the School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia.

REFERENCES

1. Cowin, S. C. (2001). *Bone mechanics handbook*. CRC press..
2. Clarke, B. (2008). Normal bone anatomy and physiology. *Clinical journal of the American Society of Nephrology*, *3*(Supplement 3), S131-S139. https://doi.org/10.2215/CJN.04151206
3. Kotha, S. P., & Guzelsu, N. (2003). Effect of bone mineral content on the tensile properties of cortical bone: experiments and theory. *Journal of Biomechanical Engineering*, *125*(6), 785-793.
4. Gibson, L. (1985). The mechanical behaviour of cancellous bone. *Journal of biomechanics*, *18*(5), 317-328.
5. Deng, L. and Yu, D. (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*. 7(3–4), 197-387.
6. Liu, F., Zhou, Z., Samsonov, A., Blankenbaker, D., Larison, W., Kanarek, A., Lian, K., Kambhampati, S. and Kijowski, R., 2018. Deep learning approach for evaluating knee MR images: achieving high diagnostic performance for cartilage lesion detection. *Radiology*, *289*(1), pp.160-169. https://doi.org/10.1148/radiol.2018172986
7. Akkus, Z., Galimzianova, A., Hoogi, A., Rubin, D.L. and Erickson, B.J., 2017. Deep learning for brain MRI segmentation: state of the art and future directions. *Journal of digital imaging*, *30*(4), pp.449-459.
8. Teller, P., & König, H. (2003). *MRI atlas of orthopedics and traumatology of the knee*. Springer Science & Business Media..
9. Liu, J., Pan, Y., Li, M., Chen, Z., Tang, L., Lu, C. and Wang, J., 2018. Applications of deep learning to mri images: a survey. *Big Data Mining and Analytics*, *1*(1), pp.1-18. https://doi.org/10.26599/BDMA.2018.9020001
10. Liu, F., Zhou, Z., Samsonov, A., Blankenbaker, D., Larison, W., Kanarek, A., Lian, K., Kambhampati, S. and Kijowski, R., 2018. Deep learning approach for evaluating knee MR images: achieving high diagnostic performance for cartilage lesion detection. *Radiology*, *289*(1), pp.160-169.
11. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Comput. Sci. 1409(1556) (2014)
12. Xiao, C., Choi, E., & Sun, J. (2018). Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*, *25*(10), 1419-1428.
13. https://colab.research.google.com/notebooks/welcome.ipynb?hl=pt

14. Erickson, B. J., Korfiatis, P., Akkus, Z., & Kline, T. L. (2017). Machine learning for medical imaging. *Radiographics*, *37*(2), 505-515.

15. Wernick, M. N., Yang, Y., Brankov, J. G., Yourganov, G., & Strother, S. C. (2010). Machine learning in medical imaging. *IEEE signal processing magazine*, *27*(4), 25-38.

16. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, *42*, 60-88.

https://doi.org/10.1016/j.media.2017.07.005

17. Ker, J., Wang, L., Rao, J., & Lim, T. (2018). Deep learning applications in medical image analysis. *Ieee Access*, *6*, 9375-9389.

18. Lundervold, A. S., & Lundervold, A. (2018). An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik*.

19. Ambellan, F., Tack, A., Ehlke, M., & Zachow, S. (2019). Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the Osteoarthritis Initiative. *Medical Image Analysis*, *52*, 109–118

20. Antony, J., McGuinness, K., O’Connor, N. E., & Moran, K. (2016). Quantifying radiographic knee osteoarthritis severity using deep convolutional neural networks. In *Pattern Recognition (ICPR), 2016 23rd International Conference on* (pp. 1195–1200). IEEE.

21. Smistad, E., Falch, T. L., Bozorgi, M., Elster, A. C., & Lindseth, F. (2015). Medical image segmentation on GPUs–A comprehensive review. *Medical image analysis*, *20*(1), 1-18.

https://doi.org/10.1109/ACCESS.2018.2874767

22. Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G. B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. *IEEE Access*, *6*, 61677-61685.

23. Fuat, “Google Colab free GPU tutorial,” 2018, [Access: 6-25-2018].[Online]. Available: https://medium.com/deep-learning-turkey/googlecolab-free-gpu-tutorial-e113627b9f5d

24. Rampasek, L., & Goldenberg, A. (2016). Tensorflow: Biology’s gateway to deep learning?. *Cell systems*, *2*(1), 12-14.

https://doi.org/10.1016/j.cels.2016.01.009