Proceedings Paper:
Zhu, Y., Green, A.C., Guo, L. et al. (2 more authors) (2020) Machine learning approaches for cancer bone segmentation from micro computed tomography images. In: Proceedings of 2020 IEEE 23rd International Conference on Information Fusion (FUSION). 2020 IEEE 23rd International Conference on Information Fusion (FUSION), 06-09 Jul 2020, Rustenburg, South Africa. IEEE , pp. 1-6. ISBN 9781728168302

10.23919/FUSION45008.2020.9190495

Machine Learning Approaches for Cancer Bone Segmentation from Micro Computed Tomography Images

Yifei Zhu*, Alanna C. Green†, Lingzhong Guo*‡, Holly R. Evans†, Lyudmila Mihaylova*
*Department of Automatic Control and Systems Engineering
†Department of Oncology and Metabolism
‡INSIGNEO Institute for in silico Medicine
The University of Sheffield, Sheffield, United Kingdom
Email: { yzhu42, a.c.green, l.guo, h.r.evans, l.s.mihaylova }@sheffield.ac.uk

***Abstract*—Many types of cancers such as multiple myeloma cause bone destruction, resulting in pain and fractures in patients and increased fatality. To quantify the degree of bone disease caused by cancer and analyse treatment response for bone repairing, accurate knowledge of the volumetry of all lesions is needed. To this end, this study proposes to apply two main approaches to the segmentation of bone lesions in cancer-induced bone disease from Micro Computed Tomography (μCT) images - structured forest-based edge detection approach and deep learning approach. A fast edge detection approach with structured forest, an extension of [1], is applied to identify the volumetry of all lesions in mice tibia, where the obtained results are evaluated against the manually labelled data, demonstrating the efficiency of the compared approaches. The Gaussian processes (Convnet GP) approach has achieved the best performance among the compared approaches, with 99.6% intersection of union and 99.7% precision. Our results demonstrate that the developed approach provides a reasonable delineation of the samples, showing the great potential towards fully automatic bone tumour segmentation.**

***Index Terms*—Machine learning, cancer bone segmentation, CNNs,FCNs, Capsule networks, Gaussian process approaches, structured forest edge-based segmentation**

I. Introduction

In the UK, one in four deaths occur due to cancer, and at this stage, the cancer has spread to bones in over 40% of patients [2]. Bone disease caused by cancer results in substantial pain, loss of mobility and fractures in patients, as well as increasing the fatality and cost of treatment [3], [4]. Unfortunately, there are no pharmacological treatments to help repair bone disease. A major limitation to developing bone-healing drugs is a lack of reliable approaches to accurately quantify bone lesions. Hence, it is essential to develop an automated approach to diagnose the bone disease accurately.

Evans et al. developed *Osteolytica* to measure cancer-induced lesions in mouse tibiae scanned by micro computed tomography (μCT) [5]. *Osteolytica* first dilates the sample bone volume image until the holes on the outer surface are filled. A contraction is then performed on the dilated volume, which stops when the contracted volume reaches the highest overlapping ratio between itself and the original volume. By subtracting the original volume and the contracted volume, the additional areas are obtained as the lesion areas, and therefore, the areas and the number of the bone lesion can be calculated [5]. The analysis using *Osteolytica* provides 0.53% average variability, which is 37 times more accurate compared with the ImageJ analysis method from [6], [7].

However, a significant problem with *Osteolytica* is that it recognises the cartilage, a normal structure in healthy and diseased bones, as a bone lesion creating a false-positive result. The cartilage can be manually excluded, but this creates a major problem when bone lesions connect with the growth plate, as the real bone lesion would also be excluded. Therefore, the aim of this study is to segment bone and cancer-induced bone lesion in three dimensions using μCT datasets. More specifically, machine learning approaches are applied on the pre-clinical CT datasets of bones with and without cancer. Two main approaches are proposed: the fast edge detection approach with structured forest, an extension of [1], and deep learning approaches, such as a convolutional neural network (CNN) [8], [9], [10], capsule network (CapsNet) [11], [12] and Gaussian process (GP) approaches [13]. Ultimately, the objective of this study is to improve the accuracy of quantifying bone lesions to facilitate reliable pre-clinical testing of new bone-targeted therapies.

The rest of the paper is structured as follows. Section II presents the developed approaches and describes the real data sets. Section III describes the performance evaluation and validation. Finally, Section IV summarises the main results.

II. Micro Computed Tomography Data and the Proposed Approach

A. Datasets

Micro Computed Tomography (μCT) datasets were used by scanning the proximal end of mouse tibiae with or without tumour [14]. The μCT datasets contain 2D transverse slices that can be rendered into a 3D dataset. Each slice has width W and height H. By aggregating N slices, the datasets are represented in 3D tensor with W×H×N. During the experiments described in later sections, we use the M9 mouse tibiae

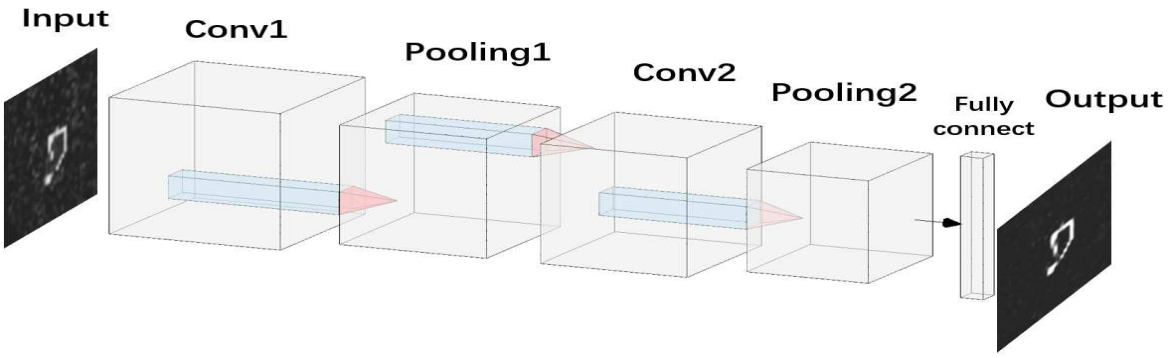


Fig. 1. CNN architecture for bone segmentation.

dataset with tumour to evaluate the approaches. M9 dataset has 1235 slices, and each slice has 1440×1440 resolution, which means that M9 dataset is a tensor with dimension of 1440×1440×1235.

B. Data preprocess

In order to segment the bone, two steps are required to be operated in order. First, the backgrounds in the image are removed; Second, boundaries of the bone are extracted.

1) Background Subtraction: In order to segment bone accurately, background subtraction is essential to enhance the bone sections. In this study, a logical mask is directly produced by binarising the slice image. Since the logical masks are the binarised images, they have the same 2D dimension as the raw images. Therefore, it is possible to correlate the raw images with logical masks.

Fig. 2 shows a sample image a) is the raw slice image. With the assistance of the logical masks, the image, b), can be generated.

2) Structured Random Forest Edge Detection: The edge detection approach applied in this study is based on the structured random forest algorithm illustrated by [1]. The main idea is to construct decision trees by training the split function of the tree. A decision tree classifies the input recursively left or right to the child nodes. Each node is associated with a split function,

$$h(x, \theta_j) \in \{0, 1\} \tag{1}$$

where θ_j is the parameter required to be optimised, and x is the input. 0 and 1 represent the two possibilities of the input, which means that the input can be split left or right to the child nodes. The input is labelled as $y \in \mathcal{Y}$ when they reach the leaves of the decision trees. The training processes become to chose a value of θ_j to maximise of the information gain I at each node.

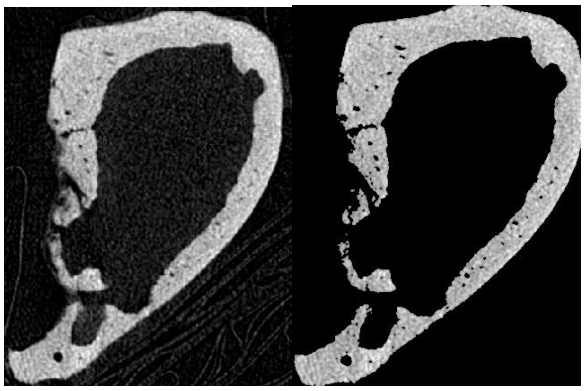
$$I_j = I\left(S_j, S_j^L, S_j^R\right) \tag{2}$$

where, $S_j \in \mathcal{X} \times \mathcal{Y}$ is the training set with $\mathcal{X}$ samples and $\mathcal{Y}$ labels. $S_j^L = \{(x, y) \in S_j | h(x, \theta_j) = 0\}$ represents to proceed on the left node and $S_j^R = S_j / S_j^L$ represents to proceeding on the right node, where $x \in \mathcal{X}$ is one of the sample.

After training the decision trees, the structured random forest architecture proposes a discrete label set $c \in \mathcal{C} = \{1, 2, ..., k\}$ that is mapped from the labels $y \in \mathcal{Y}$. The dissimilarity of the labels are approximate by calculating Euclidean distance in the intermediate mapping,

$$\Pi : \mathcal{Y} \to \mathcal{Z} \tag{3}$$

Labels with similar $z \in \mathcal{Z}$ are mapped into the same discrete label. The hierarchical label mapping labels each pixel and therefore, determines the edges. Decision trees corresponding to different feature channels, such as colour, are assembled, and the edge detection results are generated by the votes among all trees.



(a) (b)

Fig. 2. a) Raw slice image. b) Image after removing background.

C. Deep Learning Approaches

Deep learning approaches, such as convolutional neural networks (CNNs) [8], [9], are widely applied to image recognition and data forecasting. In this section, the architecture of the fully convolutional network, Capsule network and Gaussian process approaches, we applied will be introduced.

1) CNN: A CNN with three convolution layers and three pooling layers is designed. The convolution layers have 256, 128 and 64 channels, respectively, and the filter size of each layer is 3×3. A max pooling layer with a filter size of 2×2

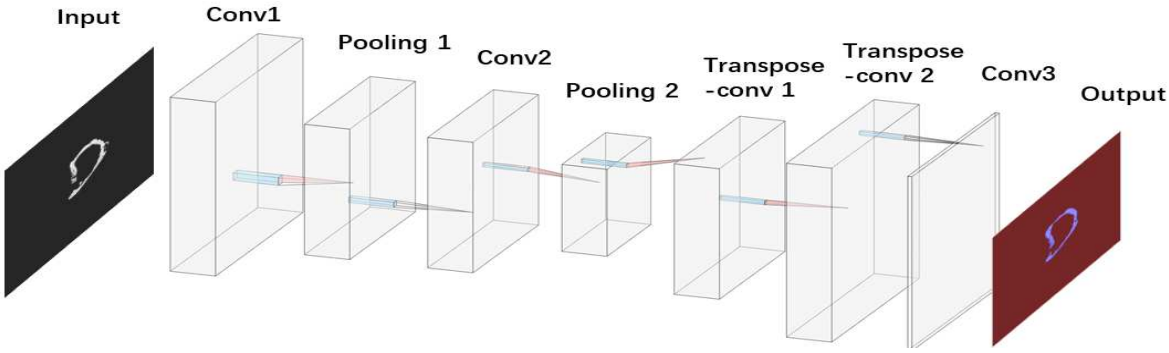


Fig. 3. Fully convolutional network architecture for bone segmentation

and a stride of 2 follows each convolution layer. All the convolution layers are activated by a Rectified Linear Unit (ReLu) activation function. At the end of the network, there is a fully connected layer. The architecture is presented in Fig. 1. The parameters of the network are presented in Table I.

TABLE I
LAYER PARAMETERS OF CNN

Layer	Parameter	Activation
Convolution1	(256,3,3)	ReLu
Polling1	(2,2)	-
Convolution2	(128,3,3)	ReLu
Polling2	(2,2)	-
Flatterning	-	-
Fully-connected	-	-

2) Fully convolutional network for segmentation:

As shown in Fig. 3, the first four layers are the same as the CNN introduced in Fig. 1. These layers performed the same job as in the CNNs, as they extracted deep feature hierarchies that encode the locations and semantics. Therefore, for pixelwise prediction and classification, the encoded information is required to connect back to pixels. Deconvolutional layers are introduced as an efficient and effective solution. Deconvolution is commonly called backward convolution, which means deconvolutional layers simply reverse the operations in convolutional layers. Therefore, deconvolutional layers achieve end-to-end learning by backpropagating the pixelwise loss [10]. Different from CNN, FCNs replace fully-connected layers, typically used for classification, by using convolutional layers to classify each pixel in the image.

Table II lists the parameters of FCN implemented for this study. The convolutional layers 1 and 2 have 32 filters with 3×3 filter size. The convolutional operations in those two layers are performed with a stride of 1 with 1 padding and activated by a ReLu. Pooling layers have filters of size 2×2 applied with a stride of 2 and 0 padding. 32 filters with 4×4 filter size applied with a stride of 1 and 1 cropping are applied in the deconvolutional layers. Convolutional layer 3 has 32 filters of size 2×2 applied with a stride of 1. The pixel classification layer applies cross-entropy as the loss function.

TABLE II
LAYER PARAMETERS OF FCN

Layer	Parameter	Activation
Convolution1	(32,3,3)	ReLu
Polling1	(2,2)	-
Convolution2	(32,3,3)	ReLu
Polling2	(2,2)	-
Deconvolution 1	(32,4,4)	ReLu
Deconvolution 1	(32,4,4)	ReLu
Convolution3	(32,2,2)	-
SoftMax	-	-
Pixel Classification	-	-

3) Capsule Network: CNNs and FCNs have shown a very good performance in different applications. However, max pooling in CNNs and FCNs lose valuable information by selecting the max values in the activations. There are only limited and pre-defined pooling mechanisms for handling variations in the spatial arrangement of data [15]. Since it is impossible to ensure that the positions of the bones are exactly the same while CT scanning, either data augmentation or image registration is required to be applied for CNNs and FCNs. CapsNet has been proposed in [11] and [12] to address the drawbacks of CNNs and FCNs. Each layer in CapsNet contains capsules that represent different characteristics of the object. The main difference between capsules and artificial neurons is that capsules are in vector forms, and their activations provide vector outputs instead of scalers in artificial neurons. More significantly, the routing algorithm updates the weights between two capsule layers, which determines the way that low-level capsules feed their input in high-level capsules. The architecture of the CapsNet implemented for this case study is presented in Fig. 4.

The convolutional layers 1 and 2 have 32 filters with 3×3 filter size. The PrimaryCaps has 128 filters with 3×3 filter size. The convolutional operations in all three layers are performed with a stride of 1 with 0 paddings and activated by a ReLu nonlinear activation. Each capsule in PrimaryCaps is an 8-dimensional vector and capsules in one cuboid share weights. The last layer is TrafficCpas that has a 16-dimensional capsule per pixel. The routing algorithm performs between PrimaryCpas and TrafficCaps with 3 iterations. The parameters of the proposed CapsNet is listed in Table III.

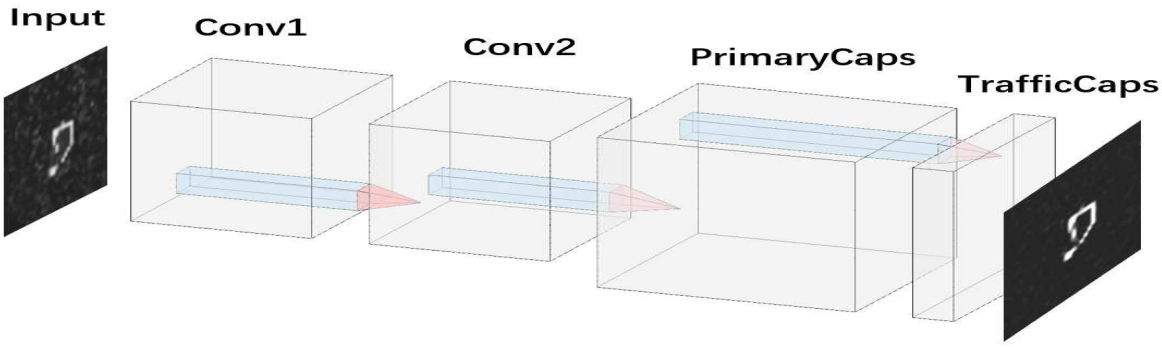


Fig. 4. Capsule network architecture for bone segmentation

TABLE III
LAYER PARAMETERS OF CAPSNET

Layer	Parameter	Activation
Convolution1	(32,3,3)	ReLu
Convolution2	(32,3,3)	ReLu
PrimaryCaps	(128,3,3) Capsule zise 8	ReLu -
TrafficCaps	Capsule size 16	-

4) Gaussian process: Nevertheless, CNNs and FCNs still face challenges, especially they are time-consuming and computationally expensive. In addition, they provide deterministic results without uncertainty analysis, and therefore, uncertainty becomes one of the hidden trouble in high-risk applications, such as biomedical applications [16]. A GP approach, one of the most powerful tools in the Bayesian inference, is potentially to equip CNNs and FCNs with the capabilities of uncertainty analysis. Garriga et al. [13] proposed that a deep CNN is essentially a shallow GP, which enabled CNN to be able to analyse the uncertainty.

A standard CNN transformation, with L hidden layers, is given as following

$$a_j^{(l+1)}(X) = b_j^l + \sum_{i=1}^{C^l} W_{j,i}^l \phi(a_i^l(X)), \tag{4}$$

where

$$X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{C^0}]^T \tag{5}$$

is the input image with height $H^{(0)}$ and width $D^{(0)}$. b_j^l is the bias and $W_{j,i}^l$ is the weight matrix that derives from the filter $U_{j,i}^l$ at the l-th layer. $\phi(a_i^l(X))$ is the activation, and $a_i^l(X)$ is the feature map from the previous layer. The elements in a filter $U_{j,i}^l$ are random, and thus, the number of potential filters could approach infinity, which means that all the filters together should average out the noise and extract the features from a polluted input. As described in [13], every element of $U_{j,i}^l$ is governed by a Gaussian distribution and the bias b_j^l is governed by another Gaussian distribution as shown in equations (6) and (7) respectively

$$u_{j,i,x,y}^l \sim \mathcal{N}(0, \frac{\sigma_w^2}{C^l}) \tag{6}$$

$$b_j^l \sim \mathcal{N}(0, \sigma_b^2). \tag{7}$$

As the weight elements and biases have a Gaussian distribution, the number of the filters, therefore, approach infinity by sampling from the corresponding Gaussian distribution. The number of filters corresponds to the number of channels in a convolutional layer. With the Central Limit Theorem (CLT), $a_j^{l+1}(X)$ subjects to a Gaussian distribution as the number of channels approach infinity.

The element-wise feature map transformation is given as,

$$A_{j,g}^{l+1}(X) = b_j^l + \sum_{i=1}^{C^l} \sum_{h=1}^{H^l D^l} W_{j,i,g,h}^l \phi(A_{i,h}^l(X)), \tag{8}$$

where C^l represents the channels. With the equation (8), we can drive the mean and covariance function,

$$\mathbb{E}[A_{j,g}^{l+1}(X)] = \mathbb{E}[b_j^l] + \sum_{i=1}^{C^l} \sum_{h=1}^{H^l D^l} \mathbb{E}[W_{j,i,g,h}^l \phi(A_{i,h}^l(X))] = 0 \tag{9}$$

$$\begin{aligned} &\mathbb{C}\Big[W_{j,i,g,h}^l \phi(A_{i,h}^l(X)), W_{j,i',g,h'}^l \phi(A_{i',h'}^l(X'))\Big] \\ &= \sigma_b^2 + \sigma_w^2 \sum_{h \in g\text{th patch}} \mathbb{E}\big[\phi(A_{i,h}^l(X)) \phi(A_{i,h}^l(X'))\big] \end{aligned} \tag{10}$$

In [13], the mean equals to 0. While the covariance function only depends on the expectation of the activation function. According to [13], the activation function is ReLu.

III. EXPERIMENT AND EVALUATION

The original μCT data are 3D, and thus, the dataset is considered as a tensor with the size of W×H×N where N donates the number of the slices, and W×H is the size of the slices. For the specific case study of M9 dataset, there are 1235 slices, and each slice has a resolution of 1440×1440. We use 2D image slices as the input.

A. Edge detection

Metrics and Evaluation: The structured random forest model [1] used in this study was pre-trained by Berkeley Segmentation Dataset and Benchmark (BSDS500) [17]. The result was evaluated by precision and recall that are formulated as following,

$$Precision = \frac{TP}{TP + FP}, \tag{11}$$

$$Recall = \frac{TP}{TP + FN}, \tag{12}$$

where TP represents true positives, FP represents false positives, and FN represents false negatives. The precision represents the percentage of the correct prediction in the total number of the prediction. The recall is the fraction of the predictions that are actually correct. Since there was a large number of pixels classified as background rather than the edge, to make the result more accurate, we only evaluated the results within specific bounding boxes that fully enclose the bone. The confusion matrix is presented in Table IV.

TABLE IV
MEAN CONFUSION MATRIX FOR FAST EDGE DETECTION

Total	Condition Positive	Condition negative
Prediction positive	8252	7532
Prediction Negative	66295	7262

By following equations (11) and (12), the mean precision is 0.529 and recall is 0.574. The visulasation is presented in Fig. 5.

B. Deep Learning Approaches

For deep learning approaches, the dataset, M9, were randomly split into 70% training and 30% testing. Due to the limitation of computational resources and efficiency, the slice images were downsampled 53 times from 1440×1440 to 27×27 for the CNN, CapsNet and Convnet GP. Only FCN was still trained with original, 1440×1440, data. The downsampling errors were evaluated by the structural similarity (SSIM). With 53-time downsampling, the SSIM equals to 0.927.

The intersection of union (IOU), given by Equation (13), the and root mean square error (RMSE) is defined as Equation (14)

$$IOU = \frac{Area\ of\ Overlap}{Area\ of\ Union}, \tag{13}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{I}(y_{x,y} - \widehat{y_{x,y}})^2}{N}}. \tag{14}$$

are used to evaluate the performance of the approaches. Here $y_{x,y}$ is ground truth pixel value and $\widehat{y_{x,y}}$ is the predicted pixel value. The CNN and CapsNet were trained with the common starting learning rate of 0.0005 and the exponential decay rate of 0.9999. The FCN was trained with a learning rate of 0.001 and the exponential decay rate of 0.9. The results were evaluated by the IOU, RMSE and precision and recall and their values are given in Table V.

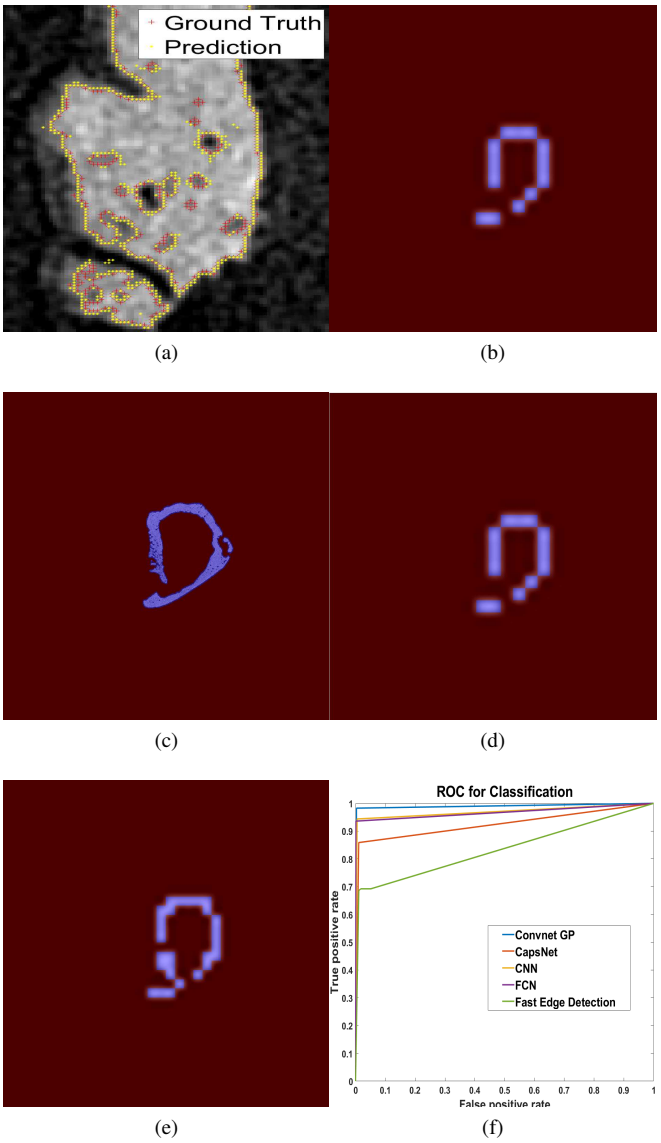


Fig. 5. a) Evaluation of the edge detection. Red represents the ground-truth points that are failed to be predicted, and yellow represents the prediction. b) is the visualisation of result obtained from CNN. c) is the visualisation of result obtained from FCN. d) is the visualisation of result obtained from CapsNet. e) is the visualisation of result obtained from Covnet GP. In b), c), d) and e), the red areas represent the background and blue areas represent the bone ares. f) ROC curve.

The Convnet GP approach has achieved the best segmentation performance with downsampled data. However, FCN achieves acceptable results with full-resolution data. On the hand of the generalization of the model, CNN, FCN and Convnet GP has their drawbacks on handling with rotation, and thus, the data augmentation is required to be applied. On the aspect of computational complexity, the applied CapsNet consumes the longest time, 62 computer hours, for training. The FCN takes second place with 31 minutes. The CNN takes 3.53 seconds, and the Convnet GP takes 2.15 seconds for

training.

TABLE V
DEEP LEARNING APPROACH EVALUATION

Approach	IOU	RMSE	Percision	Recall
CNN	99.33%	0.065	0.995	0.998
FCN(Full resoultion dataset)	86.12%	-	0.996	0.996
CapsNet	98.58%	0.086	0.989	0.996
Convnet GP	99.60%	0.031	0.997	0.999

IV. CONCLUSIONS

This paper introduces the fast edge detection and four deep learning neural networks to the segment cancer bones from the μCT datasets. The fast edge detection approach has provided 0.529 precision and 0.574 recall which are less acceptable compared with deep learning approaches. The other four deep learning NNs have provided outstanding segmentation results with a preclinical dataset. Convnet GP has achieved the highest accuracy respected either on IOU or pixel-wise evaluation (RMSE, precision and recall). However, FCN has the ability to process large scale data, and CapsNet is rotation invariant. FCN, CapsNet and Convnet GP have different strengths.

This work provides a new perspective of dealing with bone cancer segmentation and compares the effectiveness of machine learning approaches for this challenging segmentation problem. In the next stage of the research, we aim to segment the lesion area from the datasets with artificial lesions with a user-defined size to test the accuracy of our deep learning approaches in three dimensions. A challenge we face is that, while the bone is easily identifiable, the bone lesion areas are almost the same as the background. A limitation in our current approach is that it requires downsampled datasets. Since some information within the dataset is lost during downsampling, the full-size dataset will be processed in the next stage to improve the accuracy. Furthermore, the experiments have been performed on 2D slices of the μCT dataset. Given that the bone lesions are a 3D structure, it is possible that 2D CNN will not be sufficient to process the dataset. Therefore, the implementation of 3D CNN is a potential architecture that can be investigated in parallel.

V. ACKNOWLEDGMENTS

We are grateful to the support of INSIGNEO Institute for *in silico* Medicine, and to the Pump-prime fund from Department of Automatic Control & Systems Engineering and Weston Park Cancer Centre, the University of Sheffield, United Kingdom.

REFERENCES

[1] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 8, pp. 1558–1570, 2014.

[2] J. Budczies, M. von Winterfeld, F. Klauschen, M. Bockmayr, J. K. Lennerz, C. Denkert, T. Wolf, A. Warth, M. Dietel, I. Anagnostopoulos *et al.*, "The landscape of metastatic progression patterns across major human cancers," *Oncotarget*, vol. 6, no. 1, p. 570, 2015.

[3] F. Saad, A. Lipton, R. Cook, Y.-M. Chen, M. Smith, and R. Coleman, "Pathologic fractures correlate with reduced survival in patients with malignant bone disease," *Cancer*, vol. 110, no. 8, pp. 1860–1867, 2007.

[4] A. Barlev, "Payer costs for inpatient treatment of pathologic fracture, surgery to bone, and spinal cord compression among patients with multiple myeloma or bone metastasis secondary to prostate or breast cancer," *Journal of Managed Care Pharmacy*, vol. 16, no. 9, pp. 693–702, 2010.

[5] H. Evans, T. Karmakharm, M. Lawson, R. Walker, W. Harris, C. Fellows, I. Huggins, P. Richmond, and A. Chantry, "Osteolytica: An automated image analysis software package that rapidly measures cancer-induced osteolytic lesions in in vivo models with greater reproducibility compared to other commonly used methods," *Bone*, vol. 83, pp. 9–16, 2016.

[6] W. S. Rasband *et al.*, "Imagej," 1997.

[7] M. D. Abràmoff, P. J. Magalhães, and S. J. Ram, "Image processing with imagej," *Biophotonics international*, vol. 11, no. 7, pp. 36–42, 2004.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[11] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with em routing," 2018.

[12] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, 2017, pp. 3856–3866.

[13] A. Garriga-Alonso, C. E. Rasmussen, and L. Aitchison, "Deep convolutional networks as shallow Gaussian processes," *arXiv preprint arXiv:1808.05587*, 2018.

[14] A. C. Green, D. Lath, K. Hudson, B. Walkley, J. M. Down, R. Owen, H. R. Evans, J. Paton-Hough, G. C. Reilly, M. A. Lawson *et al.*, "Tgfβ inhibition stimulates collagen maturation to enhance bone repair and fracture resistance in a murine myeloma model," *Journal of Bone and Mineral Research*, vol. 34, no. 12, pp. 2311–2326, 2019.

[15] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.

[16] V. Kumar, V. Singh, P. Srijith, and A. Damianou, "Deep Gaussian processes with convolutional kernels," *arXiv preprint arXiv:1806.01655*, 2018.

[17] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2010.