Applying Ternary Net Weights to the R-Mask Neural Network to Identify Bronchopulmonary Lung Segments

N S Francis¹, N J Francis¹, S V Axyonov¹, M Saqib², Y Xu³ and S A Aljasar³

¹Engineering School of Information Technology and Robotics, Tomsk Polytechnic University, Lenin Avenue, 30, Tomsk, 634034, Russia
²Research School of Chemistry and Applied Biomedical Sciences, Tomsk Polytechnic University, Lenin Avenue, 30, Tomsk, 634034, Russia
³School of Nuclear Science and Engineering, Tomsk Polytechnic University, Lenin Avenue, 30, Tomsk, 634034, Russia

E-mail: nadinesuzannefrancis@gmail.com

Abstract. The purpose of this research is to develop an algorithm for detecting bronchopulmonary segments in lung Computer Tomography (CT) images, while reducing computational costs. The algorithm is implemented without the use of a graphics processor (GPU). The main algorithm of the proposed system introduces ternary weights into Mask R-CNN. The ternary hyperbolic tangent function replaces Mask R-CNN’s activation function to reduce overhead costs. This is a convenient and inexpensive system, designed to help radiologists to detect bronchopulmonary lung segmentation with high accuracy.

1. Introduction
Modelling and processing of medical images [1, 2] are becoming a noteworthy sector. Image enhancements and manipulation are important, as they aid radiologists to identify conflicting diseases that exist today. Considering all medical imaging techniques to date, CT imaging efficiently aids in identifying disease complexities that affect a person’s internal organs. In this research, bronchopulmonary segments of the affected lung are identified. Each bronchopulmonary segment is independent of the other, hence, identifying them not only aids in treating pathological segments, but also provide a guideline for surgery.

The previous lung segmentation methods [3] performed by various other researchers have not been yet implemented in hospitals. This is due to the inefficiency of accurately detecting pathological lung segments. Moreover, lung images are usually segmented by identifying the outer image border, nervous tissue, veins and pathologies. The existing methods provide an output rate of 43-75% in identifying bronchopulmonary lung segments over the years, as their computational accuracy and performance efficiency is low.

Since hospitals require accurate results, these existing methods need improvement. In a previous experiment, to identify bronchopulmonary segments with lesser overhead costs, various shadow regions of the lung were diagnosed, and these regions were then input into Mask R-CNN [4,5]. When the images were convolved again, they were sent into U-Net to obtain the bronchopulmonary segmented output. This method, however, needed an 8 GPU (Graphic Processing Unit) to provide the required output. Therefore, considering the requisition of a huge computational effort, this method cannot run on systems
without the required GPU specifications. Reducing the GPU makes the application accessible even on smaller devices, such as, smart devices, virtual reality, and augmented reality devices.

Besides this, the GPU pre-requisite of most CNNs eventuate because of the floating-point matrix multiplications to calculate weights and activation functions. This overhead cost can also be reduced by introducing ternary weights. These weights do not affect the performance of the system and increase the efficiency of the bronchopulmonary segments produced, as the size input into the CNN is only a single shadowed region.

The purpose of this research is to develop an algorithm for detecting bronchopulmonary segments in lung CT images using Mask R-CNN. Computational costs are reduced by replacing the floating-point weights with ternary weights in the CNN.

2. Dataset Preparation and Functioning of CNN

All medical images are obtained in a Digital Imaging and Communications in Medicine (DICOM) format and pre-processed. During pre-processing, images are resized, noise is eliminated, and they are converted to a .png or a .jpg format. Images obtained from pre-processing are then inspected to locate shadow regions representing bronchopulmonary segments. Shadow regions [6] are determined by using the Ridge based Distribution Analysis technique (RAD) [7]. This technique identifies possible structure points which define the entire lung image border. Points, which separate segments, known as ridge points, are then diagnosed. These ridge points form a histogram $\Omega(r)$ by joining their neighbouring ridge points, and can be calculated using the formula:

$$A(r, \sigma) = T(r, \sigma_s) \times (\nabla \Omega(r, \sigma_g) \cdot \nabla k(r, \sigma_g))$$

where, $r$ is the central ridge point inside an area $A$, $\sigma_s$ is the neighbourhood range [8] that forms the segment of the lung image [8], the value of $\sigma = \{\sigma_s, \sigma_g\}$, $T$ is the total number of ridge points that forms a single segment, and $k$ is the gaussian kernel to calculate the vector field calculus.

A single histogram marks a single shadowed segmented region. In this way, all the shadow regions representing possible bronchopulmonary segments are identified.

Each identified shadow segment is then sent separately into Mask R-CNN. The functioning of the CNN is divided into two phases: a) Region Proposal Network (RPN) and b) Feature Extraction and ROI Pool.

2.1. Region Proposal Network (RPN)

In the RPN [9] phase as seen in fig 1, the shadow segmented image provides a set of output proposals along with a score. This score represents the accuracy of the output proposal. These proposals and scores are collected by passing a sliding window over the convolved image. Each output proposal and score is then stored in a bounding box, which when collected, are further stored in an ROI pool.

![Figure 1](image.png)

**Figure 1.** Capturing output proposals from RPN (i) Segmented CT input that is convolved (ii) Sliding window to select segmented region (iii) Sliding Window regions stored in a Bounding box (iv) Convolved images stored in an ROI pool.

2.2. Feature Extraction and ROI pool
In this phase, features are extracted from each bounding box of the ROI pool [9, 10] by classification and regression, the result of which is a box offset and class. In parallel to this operation, binary masks are identified for each region of interest (ROI). A feature map is created with each ROI, and inserted into the ROI pool. To eliminate the excess overhead cost, ternary weights are quantised to Mask R-CNN instead of floating-point weights, and are divided into spatial bins [11]. Max pooling is then applied to these bins. Finally, the processes of quantisation [12] and bilinear interpolation are performed on the pooled output to display the bronchopulmonary segmented result. Fig 2 displays how features are extracted from the ROI pool and also the working of ternary weights to identify bronchopulmonary lung segments.

![Diagram](image)

**Figure 2.** Overall Working Structure (i) Phase I – Region Proposal Network (RPN) that convolves and stores output in the ROI Pool (ii) Phase II - Feature Extraction with Mask R-CNN by reducing the overhead using Ternary weights.

### 3. Calculation of Ternary Weights and Activation Function

To calculate the ternary weight [13] and activation functions of CNN, the precision of weights allotted to various layers of the CNN is reduced by rounding off floating-point weights. This process involves multiplying a floating-point scaling parameter with the floating-point weights to produce an absolute weight value. The scaling parameter $\alpha$ is calculated so that the ternary weights ($T_w$) lies within a finite series. To avoid discrete optimization [13,14] which is time consuming, $|T_w|$ is determined from a normal distribution $D(0, \sigma^2)$, so that $\alpha$ always lies within a positive range. This range is kept positive by multiplying $\alpha$ with a constant ($0.6\sigma/x = 0.75/x$). This constant is approximated to $0.7/x$ according to the rule of thumb. Hence, $\alpha$ is calculated for all weights $w$, from 1 to $x$, using the equation:

$$\alpha = 0.7/x \sum |T_w|$$

The scaling parameter produces ternary values of $-1$, 0, or $+1$. The values are calculated based on Hamming distances, such that the ternary weight value $T_w \approx \alpha T_w$. Since hamming distance is usually used for calculating the distance between binary codes, it is a good variant to the Euclidean distance used in the original mask R-CNN algorithm as it reduces the computational overhead. The ternary weights $T_w$ is $+1$ for all values greater than $\alpha$, 0 for values lesser than or equal to $\alpha$, and $-1$ for all other values.

While training the network, when the weights are quantised to R-Mask CNN, full precision quantisation values are calculated. This is done to reverse individual floating-point values [15] to their equivalent full value.
Ternary hyperbolic tangent function is used during the hidden layer convolution process, to calculate the actual ternary values within the intermediate layers. Unlike ternary weights [15, 16], this activation function does not require a constant, as it adapts and produces whole values for the intermediate layers. The ternary hyperbolic tangent function \( \tanh_T \) is calculated using the formula:

\[
\tanh_T = 0.5(2\delta y - \delta) - 0.5(-2\delta y - \delta)
\]

where, \((2\delta y - \delta)\) and \((-2\delta y - \delta)\) are the hyperbolic tangents used for calculating the input values of intermediate layers. The constant \( \alpha = 0.5 \) is used for the calculation of all hidden weights.

The combination of ternary weights and activation functions help in removing floating-point calculations within the CNN. It reduces overhead costs [16] involved, and memory required to accurately identify bronchopulmonary lung segments.

4. Dataset training and testing the results

The network is trained to consist of around 400 iterations with about 50 epochs each. The initial learning rate of the CNN is approximately 0.0030, and the cross-entropy loss for the pre-processed bitmap image is about 2.4 %. Approximately 249 lung images are used for training, and around 120 lung images for testing. The training is divided into 6 samples. Each of the first 3 training samples contain 41 scans, and each of the latter 3 training samples contain 42 scans. These lung scans are obtained from local medical hospitals in the DICOM format.

The datasets are trained with 3 models, namely Mask R-CNN, Mask R-CNN with U-Net, and Mask R-CNN with Ternary Weights, to check the results and the overhead cost of each neural network. The samples undergo a Wilcoxon signed rank two phased hypothesis test to check the performance and functionality of the system. Memory used to perform bronchopulmonary segmentation is recorded and displayed along with the output performance. Test results recorded in Table 1 depict that Mask R-CNN provides 10% better segmentation results and uses lesser memory space when compared to other GPU models. The segment dice (%) gives the mean of the segment accuracy. The segment error can be calculated using the formula, mean ± Standard Deviation. This error calculation determines the confidence interval, which in turn detects the accuracy of the identified bronchopulmonary segments.

| CNN Model                              | Segment dice, % | Standard Dev | P-value | Used memory, Mb |
|----------------------------------------|-----------------|--------------|---------|-----------------|
| Mask R-CNN (Full Resolution)           | 64.70           | ±11.4        | 0.001   | 15.6            |
| Mask R-CNN with U-Net                  | 76.80           | ±10.6        | < 0.001 | 10.8            |
| Mask R-CNN (Ternary Weights)           | 86.87           | 9.4          | < 0.001 | 0.78            |

To analyse bronchopulmonary segmentation results from the 3 CNN models, sensitivity and specificity are calculated for all the data samples.

| Architecture                        | Data Samples, S | Sensitivity | Specificity |
|-------------------------------------|-----------------|-------------|-------------|
| S1                                  | 0.7895          | 0.5455      |
| S2                                  | 0.7692          | 0.4667      |
| S3                                  | 0.6842          | 0.7273      |
| S4                                  | 0.6154          | 0.6250      |
| S5                                  | 0.7826          | 0.5263      |
| S6                                  | 0.6250          | 0.6923      |
| S1                                  | 0.8571          | 0.6316      |
| S2                                  | 0.8519          | 0.6429      |
| S3                                  | 0.8333          | 0.8182      |
|   |      |      |
|---|------|------|
| S4 | 0.7083 | 0.7222 |
| S5 | 0.8400 | 0.6471 |
| S6 | 0.6875 | 0.7308 |

|   |      |      |
|---|------|------|
| S1 | 0.9565 | 0.8000 |
| S2 | 0.8929 | 0.7857 |
| S3 | 0.9412 | 0.8571 |
| S4 | 0.8696 | 0.8947 |
| S5 | 0.9600 | 0.8235 |
| S6 | 0.7500 | 0.7692 |

The graph in Figure 3, aids in analysing the efficiency of detected bronchopulmonary segments during each epoch of the training phase for the three models. The graph also depicts the loss of pixels, if any. The sparsity of the image during training remains constant for a full resolution Mask R-CNN, but increases the zero value for the other CNN modules. A constant rise of the slope in the graph depicts continuity in training the module, whereas a stagnant slope depicts the same values obtained from every epoch.

![Graph depicting continuity of dataset training.](image)

From the above experimental calculations and graphical analysis, the proposed model indicates that it is far more efficient in identifying bronchopulmonary lung segments without using a GPU. Overhead costs are also reduced.

5. Conclusion
The existing CNNs used for identifying bronchopulmonary lung segments in CT images usually required huge GPU processors, which were time consuming and expensive. The successful method proposed, reduces the processing time taken and can be used in future for medical related imaging. Furthermore, the system can be implemented in smaller devices such as smartphones, as the amount of space consumed during the entire process is minimal. The Mask R-CNN incorporated with ternary weights, serves as an advantage to identify bronchopulmonary lung segments efficiently and accurately.
References

[1] Tajbaksh N, Shin J Y, Gurudu S R, et al. 2016 Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? IEEE Transactions on Medical Imaging. 35 (5) 1299-12

[2] Zheng Li and Le Yiran 2018 A Review of Image Segmentation Methods for Lung Nodule Detection Based on Computed Tomography Images. MATEC Web of Conferences, 232

[3] Hesamian M H, Jia W, He X et al. 2019 Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges. J Digit Imaging 32 582–596

[4] Amanda A R and Widita R 2016 Comparison of image segmentation of lungs using methods: connected threshold, neighborhood connected, and threshold level set segmentation. Journal of Physics: Conference Series 694 012048

[5] He K, Gkioxari G, Dollár P & Girshick R B 2017 Mask R-CNN. 2017 IEEE International Conference on Computer Vision (ICCV), pp 2980-88

[6] Kim D S, Arsalan M & Park K R 2018 Convolutional Neural Network-Based Shadow Detection in Images Using Visible Light Camera Sensor. Sensors (Basel, Switzerland) 18 (4) 960

[7] Hou L, Samaras D, Kurc T M, Gao Y, Davis J E & Saltz J H 2016 Patch-based Convolutional Neural Network for Whole Slide Tissue Image Classification. Proceedings. IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp 2424–33

[8] Zhang S, Wu R, Xu K, Wang J and Sun W 2019 R-CNN-Based Ship Detection from High Resolution Remote Sensing Imagery. Remote Sens 11 631

[9] Ren S, He K, Girshick R & Sun J 2017 Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39 (6)1137–49.

[10] Qin Y, He S, Zhao Y & Gong Y 2016 RoI Pooling Based Fast Multi-Domain Convolutional Neural Networks for Visual Tracking. Proceedings of the 2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016). doi:10.2991/aiie-16.2016.46.

[11] Xie B, Zhang H K & Xue J 2019 Deep Convolutional Neural Network for Mapping Smallholder Agriculture Using High Spatial Resolution Satellite Image. Sensors (Basel, Switzerland) 19 (10) 2398

[12] Wei X, Liu W, Chen L, Ma L, Chen H & Zhuang Y 2019 FPGA-Based Hybrid-Type Implementation of Quantized Neural Networks for Remote Sensing Applications. Sensors (Basel, Switzerland) 19 (4) 924

[13] Li F and Liu B 2016 Ternary weight networks. CoRR abs/1605.04711.

[14] Natzina Juanita S R F et al. 2020 Analysis and Detection of Community Acquired Pneumonia Using PSPNET with Complex Daubechies Wavelets. Indian Journal of Computer Science and Engineering 11 (3) 217-226

[15] Zheng S, Liu Y, Yin S, Liu L and Wei S 2018 An efficient kernel transformation architecture for binary- and ternary-weight neural network inference. In Proceedings of the 55th Annual Design Automation Conference (DAC ’18). Association for Computing Machinery, 137 1–6 DOI:https://doi.org/10.1145/3195970.3195988.

[16] Heinrich M P, Blendowski M and Oktay O 2018 TernaryNet: faster deep model inference without GPUs for medical 3D segmentation using sparse and binary convolutions. International Journal of Computer Assisted Radiology and Surgery 13 (9) 1311-20