# A QuadTree Image Representation for Computational Pathology

Robert Jewsbury Abhir Bhalerao Nasir Rajpoot

TIA Centre, Department of Computer Science, University of Warwick, UK

{rob.jewsbury,abhir.bhalerao,n.m.rajpoot}@warwick.ac.uk

## Abstract

*The field of computational pathology presents many challenges for computer vision algorithms due to the sheer size of pathology images. Histopathology images are large and need to be split up into image tiles or patches so modern convolutional neural networks (CNNs) can process them. In this work, we present a method to generate an interpretable image representation of computational pathology images using quadtrees and a pipeline to use these representations for highly accurate downstream classification. To the best of our knowledge, this is the first attempt to use quadtrees for pathology image data. We show it is highly accurate, able to achieve as good results as the currently widely adopted tissue mask patch extraction methods all while using over 38% less data.*

## 1. Introduction

Recent advances in digital pathology and the adoption of machine learning methods for computer vision have resulted in many efforts to develop deep learning algorithms for the analysis of information-rich pathology images. The tissue slides analysed by pathologists are very heterogeneous for many reasons including the variety of tissue types, differences in staining protocols and presence of artifacts such as smudges or out of focus regions, to name only a few [21]. Furthermore, the images are scanned at high resolution, frequently containing hundreds of millions of pixels. Whole Slide Images (WSIs) present a unique challenge in that most state-of-the-art computer vision systems rely on convolutional neural networks (CNNs) [14] and are unable to process images larger than a few hundred pixels square in size at once. As such, existing supervised learning methods to analyse large histopathology images typically break them up into small patches losing contextual, global information contained in the larger field of view (FoV) of the entire image [8, 10]. Another major limitation of these patch-based methods is that they feed every tissue patch into a CNN, whether it is useful or not.

### 1.1. Representation learning in computational pathology

There has been comparatively little research into novel ways of representing pixel data in pathology images. The vast majority of research has focused on using the patch-based paradigm for a huge variety of tasks from classification through to tumour and tissue segmentation and cell counting regression problems [15]. Tellez *et al.* [24] propose a two-step method to use CNNs for analysis of gigapixel images using only the image level label termed Neural Image Compression (NIC). They split the input image into adjacent, non-overlapping patches and train an unsupervised compression network to encode semantic information from each patch into a lower-dimensional feature vector. They arrange the extracted feature vectors with the same spatial correspondence as in the original WSI and train a CNN classifier on the extracted feature representation to predict the image level label. With this they are able to achieve an image-level performance only 7% different compared to a fully supervised baseline. However, the process of creating a representation of the WSI still breaks the image up into patches and assumes the patch level encoder network is able to retain the global information between patches in the generated feature representation. Furthermore, as it is a deep feature representation, it is not easily interpretable and there is no way of verifying whether the extracted feature vectors retain the global information they share with other feature vectors.

Instead of using a neural network to encode adjacent regions of a histopathology image, we propose to use quadtrees built in a top down fashion from the entire image [7]. Quadtrees are tree data structures, historically used in image compression [23]. They are constructed by recursively partitioning a two-dimensional space into four subregions of equal size and storing the information of each

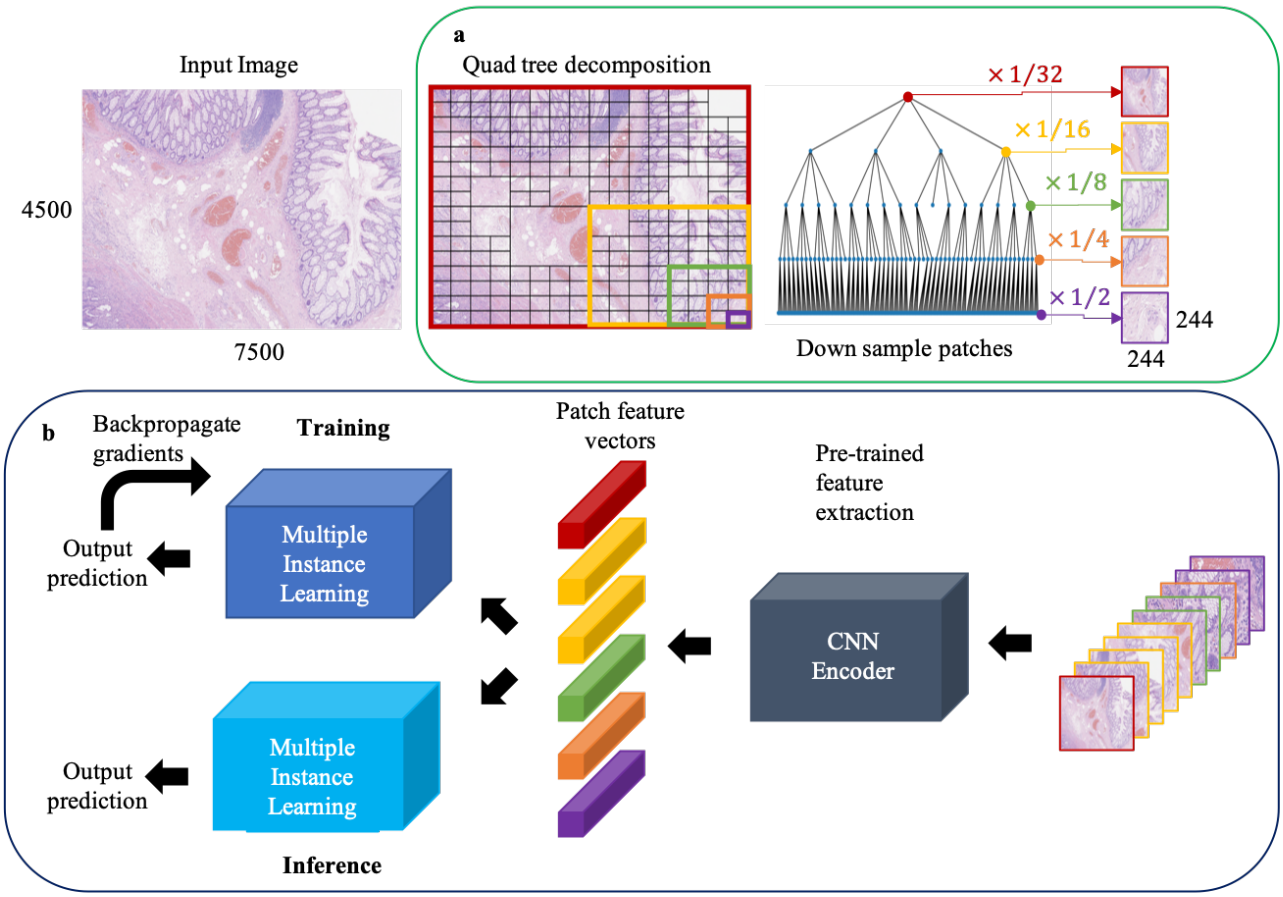


Figure 1. **Overview of our framework. a** A quadtree is constructed from a given input image, image regions are extracted from each node of the quadtree and down sampled to be of size $244 \times 244$ pixels and stored as a bag of patches. **b** Patches are passed through a feature network. We used a ResNet18 [9] pre-trained on ImageNet [5] for all our experiments to encode the feature vectors. These feature vectors are then passed through a MIL framework to generate the final output prediction.

region in a node within a tree. Each node contains information about its corresponding region and has either exactly four children or none in the case of leaf nodes. As such, the tree data structure can be used to represent the information contained in the 2D space in a more compressed, data efficient way. By traversing the tree, the information from the original space can be extracted. The depth of the quadtree can depend on the size of the input image and the distribution of information within the space that is being decomposed. If there is sufficient relevant information within a given region, the algorithm will continue adding nodes to the tree and extending its depth until this no longer holds. The maximum depth of the tree can be limited by truncating the growth at a user defined depth.

Quadtrees have not been used extensively within machine learning. The three-dimensional analogue, octrees, have been explored for tasks such as shape and scene completion with voxel arrays [25, 26]. Wang *et al.* [25] use an Octree-based Convolutional Neural Network (O-CNN) for 3D shape analysis. By building an octree representation of 3D shapes and restricting the computation to the leaf nodes they are able to more efficiently store the information and achieve comparable performance with existing methods while using less memory. Jayaraman *et al.* [12] applied this idea with Quadtree Convolutional Neural Network (QCNN) to sparse two-dimensional handwriting data sets resulting in more efficient memory usage and computation time compared to a standard CNN.

However, Wang *et al.*'s [25, 26] and Jayaraman *et al.*'s [12] work solely uses binary data. Wang *et al*'s O-CNN works with positional data while Jayaraman *et al*'s QCNN uses greyscale image data. Histopathology data is 2D RGB colour data. In this work, we propose to use the quadtree itself as an image representation while taking advantage of

the compression benefits of quadtrees by applying them to RGB data. To our knowledge this is the first time quadtrees have been used for this purpose.

### 1.2. Quadtree image representation

We propose a new image representation built with computational pathology in mind using a type of tree data structure called a quadtree. Unlike existing digital pathology representation methods our framework uses an image representation and is not necessarily a feature representation. We use this quadtree representation to extract patches at varied resolutions in different regions of the images. This leads to greater performance in downstream tasks compared to existing patch extraction techniques all while using less data due to the quadtree construction. We show that our method is able to identify significant regions within an image relevant to clinical diagnosis and generate an interpretable, tree structure representation.

### 1.3. Contributions

Our contributions can be summarised as follows:

1. We propose an image representation for computational pathology images and a pipeline able to predict image level labels using a single GPU.

2. We compare several methods for constructing the quadtree representation using different colour spaces and information quantification functions.

3. We evaluate our pipeline on a histopathology colorectal adenocarcinoma (CRA) data set used to train the system to classify an image as cancerous or non-cancerous.

4. We generate attention heatmaps to discover which regions of the image the model found significant in predicting the image level label.

The remainder of the paper is organised as follows: Section 2 explains our method in depth; Section 3 details our experimental results; our discussion and conclusions from our results are stated in Sections 4 and 5 respectively.

## 2. The Quadtree Framework

In this section, we present our method for constructing quadtrees from histopathology images and how we use them for downstream analysis. Our pipeline method consists of two main stages.

For a given image, a quadtree is constructed and patches extracted from the tree's nodes. These patches are then treated as a bag of instances and used in a multiple instance learning (MIL) paradigm to generate the image level prediction [6].

### 2.1. Building quadtrees

A quadtree represents a partition of a two-dimensional space obtained by recursively decomposing the region into four equal quadrants and sub-quadrants where each node in the tree contains information corresponding to a given specific partition of the original space [7]. In our case the two-dimensional space is that of an RGB image. The quadtree algorithm works on the intuition that if an image or a sub-region within an image, represented by a node in the quadtree, contains sufficient "interesting" information it should be divided further into four equally sized sub regions. If this occurs, the tree is expanded by adding four child nodes to the original node being evaluated. However, if a region does not contain sufficient interesting information, then no child nodes are added and the node is made a leaf node. When all regions have been decomposed down to leaf nodes, *i.e.* they do not need to be split any further, then the quadtree is complete. An example of the quadtree representation is shown in Figure 1a including an example of a decomposed image and its quadtree representation.

**Algorithm 1:** Quadtree construction algorithm QuadTree

**Input :** Image: $i \in \mathbb{R}^{m \times n \times 3}$, threshold: $t$, tree depth: $d$, criterion: $c$, quadtree: $Q =$ **null**
**Output:** quadtree $Q$
add node $n$ to $Q$ at depth $d$
**if** $n$ *has a parent* **then**
  connect parent to $n$
**end**
**if** $c(i) \leq t$ **or** *depth of* $Q = d$ **then**
  return $Q$
**else**
  split $i$ into 4 subquadrants of equal size $[i_1, i_2, i_3, i_4]$
  **for** $j$ *in* $[i_1, i_2, i_3, i_4]$ **do**
    QuadTree($j$)
  **end**
**end**

Mathematically, for use with RGB images we represent this idea with the following components:

1. A function to quantify the information within a given region which we refer to as the *criterion*;

2. A threshold value which determines the amount of information a region needs to contain to justify splitting the region further.

These components allow us to compute whether the splitting process should occur in a region given the criterion and splitting threshold.

For example, let us say we have a given region $i \in \mathbb{R}^{m \times n \times 3}$, a splitting threshold $t$ and a criterion function $c$. We can compute $c(i)$ and then compare it with the threshold $t$. If $c(i) > t$ then this means the given region has enough information to justify splitting the region further and expanding the overall quadtree. Conversely, if $c(i) \leq t$ then the region does not have sufficient information to warrant expanding the quadtree and the given node of the tree becomes a leaf node.

#### 2.1.1 Splitting criteria

We explored several different splitting criterion functions. We used the entropy [22] and mean pixel value of an image region as a measure of the amount of significant information in a region. Furthermore, we used these criteria with the images in the native RGB space and two additional colour spaces. All images we used were stained with Haematoxylin and Eosin (H&E). The haematoxylin dye stains basophilic tissue structures blue, in particular nuclei are heavily stained while cytoplasmic tissue regions are lightly stained. We hypothesise that by focusing on the bluer regions of the image, this will guide the quadtree splitting method to focus on the nuclei within the images which are usually the main feature of interest for most downstream tasks. As such we explored converting the images into blue ratio space [16] to highlight the areas dense in nuclei. We also used colour deconvolution [20] to separate out the different stains as colour channels and used the Haematoxylin channel as a proxy for the number of cells.

#### 2.1.2 Determining the splitting threshold

The splitting threshold $t$ used in the quadtree algorithm is a hyperparameter which we determined from the data set. For a given criterion function and colour space, such as entropy in RGB space, we computed the value of this criterion for each image in the data set. We then calculated the mean $(\mu)$ and standard deviation $(\sigma)$ of this distribution and used these values to determine the splitting threshold. We performed an ablation study for this threshold and how it affects both qualitative and downstream performance, as described in Section 3.1.

#### 2.1.3 Learning from quadtrees

In each node of the quadtree we stored a down-sampled copy of the original image region which corresponds to that node's position within the quadtree decomposition. We used image interpolation to down-sample the image regions to all be of size $244 \times 244$ for later processing by a pre-trained CNN for feature extraction. This allows us to store patches at different scales from the original image. This incorporates the wider FoV present in the patches corresponding to shallow nodes in the tree while the deeper nodes contain the fine grained detail at high magnification.

### 2.2. MIL framework

Having generated a collection of down-sampled image regions from the original image using our quadtree method we treated this collection as a "bag of instances" or patches. This lends itself to the Multiple Instance Learning (MIL) paradigm, a weakly supervised learning approach, where the bag of instances has a single label but instance level labels are not available [6]. The label of the bag is positive $(Y = 1)$ if at least one instance within the bag is positive and conversely the overall bag label is negative $(Y = 0)$ if no instance within the bag is positive.

Formally, for a given image $X \in \mathbb{R}^{m \times n \times 3}$ with associated label $Y \in \{0, 1\}$ we create a bag $B$ of instances:

$$B = \{(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)\}, \tag{1}$$

where $x \in \mathbb{R}^{244 \times 244 \times 3}, y \in \{0, 1\}$. The instance level labels $y_n$ are hidden to the learner. To classify this bag of images, we can use the following function

$$\Theta(X) = g(\alpha(f(x_1), ..., f(x_k))), \tag{2}$$

where $f$ is a transformation of instances $x_k$ to a lower-dimensional embedding, $\alpha$ is a permutation-invariant aggregation function and $g$ is a transformation to generate the overall class probabilities for the bag.

There are two main approaches to MIL:

1. *Instance-based*: Here $f$ classifies each instance individually, assigning a class label for each instance and then $\alpha$ combines these predictions together to generate an overall prediction for the bag using operations such as mean, maximum etc [4]. Finally, $g$ is the identity function in this case.

2. *Embedding-based*: Instead of classifying each instance individually, here $f$ maps each instance to a lower-dimensional embedding, $\alpha$ then obtains a bag representation independent of the number of instances in the bag and $g$ classifies these bag representations to obtain the overall prediction [18].

We initially tested the instance-based method but found the classifier was difficult to train to a sufficient accuracy causing poor performance. The embedding-based approach has less bias than the instance-based one and was found to perform better, hence was used for our experiments. For a bag of image patches we used a ResNet-18 pre-trained on ImageNet to encode each image patch as a feature vector turning the bag of image patches into a bag of feature vectors.

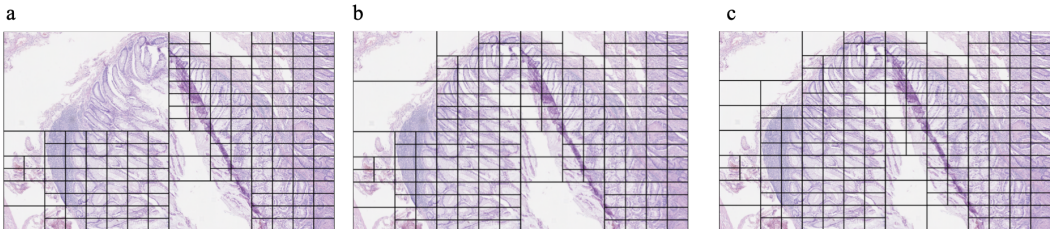


Figure 2. **Example of quadtree decompositions in different colour spaces**. **a** Image decomposed using entropy in RGB space. **b** Image decomposed using mean pixel value in blue ratio space. **c** Image decomposed using mean pixel value of haematoxylin channel. The splitting threshold was set to the mean ($\mu$) minus one standard deviation ($\sigma$) for each case.

#### 2.2.1 MIL methods

We used two different MIL methods based on an attention mechanism [3]. First, we used Attention-based MIL (AMIL) [11]. Attention-based MIL modifies the embedding-based MIL approach by changing the MIL pooling operation, $\alpha$, where for a bag $B = \{h_1, ..., h_k\}$ with $K$ embeddings the overall prediction is obtained by

$$\boldsymbol{z} = \sum_{k=1}^{K} a_k \boldsymbol{h_k}, \tag{3}$$

where

$$a_k = \frac{\exp\{\boldsymbol{w}^T \tanh \boldsymbol{V}{\boldsymbol{h_k}}^T\}}{\sum_{j=1}^{K} \exp\{\boldsymbol{w}^T \tanh \boldsymbol{V}\boldsymbol{h}_j^T\}}, \tag{4}$$

$\boldsymbol{w} \in \mathbb{R}^{L\times 1}$ and $\boldsymbol{V} \in \mathbb{R}^{L\times M}$ are parameters. For full details we refer the reader to Ilse *et al.*'s original paper [11].

We also explored a recent expansion of Ilse *et al.*'s Attention-based MIL. Lu *et al.*'s Clustering-constrained Attention Multiple Instance Learning (CLAM) was developed with computational pathology in mind [17]. In CLAM, the attention network predicts a set of attention scores for each class in the classification problem. Lu *et al.* use the same attention backbone in the first two layers of the network as Ilse *et al.* but then split the network into $n$ parallel attention branches in an $n$-class classification problem along with an additional instance-level clustering layer for each class to obtain the overall prediction. Again, we refer the reader to Lu *et al.*'s original paper for full details of the method [17].

## 3. Experimental Results

To evaluate our proposed method we used the CRAG data set from Awan *et al.* [2]. This contains 139 non-overlapping images extracted from 38 digitised WSIs of colorectal adenocarcinoma (CRA) at 20× magnification. The images varied slightly but on average were around $4500 \times 7500$ pixels in size. Of the 139 images, 71 were classified as normal tissue, 33 as low grade and 35 as high grade. We merged the low grade and high grade classes into a cancerous class to create a binary classification problem suitable for the MIL paradigm. In total, we had 71 images in the non-cancerous or negative class and 68 cancerous or positive images creating a fairly balanced classification problem. Awan *et al.* also provided the training/validation split they used for 3-fold cross validation so a fair comparison between our results and theirs could be drawn.

Given the size of the original images, we set the maximum depth $d$ of our quadtrees to be $4$. This was done because if we allowed the trees to create nodes at a deeper level then the original image regions each node represents would be smaller in size than the $244 \times 244$ patch size that are fed into pre-trained networks. Upsampling would have been required to fix this and it would have been inconsistent with the process used on the nodes at every other level within the tree. We kept the trees at depth $4$ for all of our experiments.

During training, we normalised each patch using the mean and standard deviation of ImageNet, augmented each extracted patch using random horizontal and vertical flips with probability 0.5 as well as small, random adjustments to the brightness, contrast, saturation and hue of the input images. We used the Adam optimiser [13] with a loss rate of $5\mathrm{e}{-4}$, betas of $0.99$ and $0.999$ and weight decay of $1\mathrm{e}{-4}$. All experiments were performed on a single Nvidia Quadro RTX 5000 GPU.

### 3.1. Quadtree decompositions

To evaluate the different proposed splitting methods for the quadtrees described in Section 2.1.1 we performed a qualitative analysis of all 139 images' quadtree decompositions under each method. For each method, we tested a variety of thresholds but kept them consistent in our comparisons. For example, we would only compare images that had been split where the threshold was set to be the same e.g. mean ($\mu$) minus 1 standard deviation ($\sigma$) for each given colour space and method with images split at the same threshold, not with decompositions where the threshold had been set to $\mu - 1.25\sigma$ or any other level.

We found that the mean pixel value in the Haematoxylin channel gave the best qualitative decomposition of the in-

| Method | Accuracy | AUROC |
|---|---|---|
| All patches AMIL | $87.83 \pm 7.97$ | $0.93 \pm 0.06$ |
| Blue ratio AMIL | $77.01 \pm 6.77$ | $0.94 \pm 0.06$ |
| RGB AMIL | $90.72 \pm 8.82$ | $0.96 \pm 0.04$ |
| Haematoxylin AMIL | $94.25 \pm 3.31$ | $0.98 \pm 0.02$ |
| All patches CLAM | $88.55 \pm 8.60$ | $0.94 \pm 0.06$ |
| Blue ratio CLAM | $80.54 \pm 5.88$ | $0.93 \pm 0.04$ |
| RGB CLAM | $92.06 \pm 8.25$ | $0.98 \pm 0.02$ |
| Haematoxylin CLAM | $\mathbf{97.13 \pm 1.21}$ | $\mathbf{0.98 \pm 0.02}$ |

Table 1. Comparison of Attention MIL (AMIL) and CLAM models trained on extracted patches from the 3 different colour spaces at a threshold of $\mu - \sigma$ and with all possible non-overlapping patches from the image. Results reported are *mean* $\pm$ *standard deviation*, the variation in performance occurs due to differences in cross validation performance

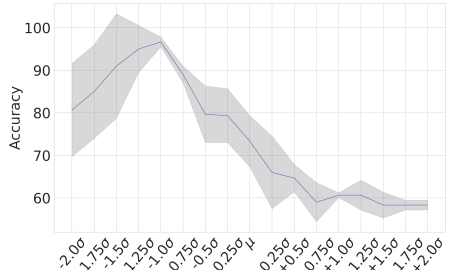


Figure 3. Accuracy of a CLAM model trained on data sets extracted at different thresholds ranging from $\mu - 2\sigma$ to $\mu + 2\sigma$ in the Haematoxylin channel using the mean pixel value. Grey area indicates $\pm$ one standard deviation in the model's performance across 3 fold cross validation.

put images. Figure 2 shows an example of this. The method consistently split the image into finer grained detail in the presence of tissue while ignoring the background or less informative regions such as empty space or large areas of adipose tissue. Furthermore, it would aggressively split regions in the presence of cancerous tissue while with non cancerous tissue the method would sometimes only split down to a depth of 2 or 3 in more homogeneous areas such as large regions of connective tissue. This was not found to be the case in large regions of cancerous tissue where the method would consistently split the image down to the maximum allowed depth. Additionally, the threshold of $\mu - \sigma$ was empirically found to give the best qualitative splits by consistently following and splitting the tissue regions within the images.

We trained the Attention MIL [11] and CLAM [17] models for 20 epochs using the patches extracted with each different method. As a baseline control comparison, we also divided every image into non-overlapping patches of size $244 \times 244$ denoted as the "all patches" baseline. The results in Table 1 show that our qualitative findings that the Haematoxylin channel method produced the best decompositions of the images also corresponded to the best downstream model performance with both Attention MIL and CLAM when compared to the other methods with all other variables kept constant. Furthermore, two of the three colour spaces, RGB and Haematoxylin channel, outperformed our baseline of using every possible patch in an image as a bag of instances. The third colour space, blue ratio space, performed relatively poorly as it almost always predicted the positive class giving a very high sensitivity but leading to overall worse performance than the other methods. Additionally, it was found that in each respective colour space and with the all patches baseline the CLAM model outperformed the Attention MIL model in terms of average performance with a smaller standard deviation for all cases except the all patches baseline.

### 3.2. Ablation study

We performed an ablation study of the splitting thresholds used with the best performing approach, using the mean pixel value in the Haematoxylin channel. We varied the threshold from $\mu - 2\sigma$ to $\mu + 2\sigma$ in steps of $0.25\sigma$ and trained a CLAM model with 3 fold cross validation at each of the 17 thresholds, all other hyperparameters were kept constant. We found the model had the greatest performance using the best qualitative threshold of $\mu - \sigma$ as it had the highest average performance and the smallest standard deviation across cross validation folds as shown in Figure 3.

### 3.3. Comparisons with other methods

We performed an additional comparison using the standard practice for extracting patches in tissue regions in computational pathology. To select only regions with tissue we thresholded an image's intensity to separate the tissue from the background and created a tissue mask with Otsu's method [19]. We obtained a set of locations within the tissue area to extract patches from with the super-pixel algorithm such that the locations covered the entire tissue region [1]. To explore the significance of the non-leaf nodes with the quadtree structure in downstream tasks we also compared performance using just the leaf nodes patches extracted by the Haematoxylin channel method.

Table 2 shows that the patches extracted by our Haematoxylin channel quadtree method yield as good performance as the segmented tissue patches with a smaller standard deviation between cross validation folds. In total the segmentation method extracted 52,029 patches, 374 per image, while the Haematoxylin channel quadtree method extracted 32,099, 231 per image, a 38.31% reduction in data which still yielded as good if not better final model performance with all other training hyperparameters the same.

| Method | Accuracy | AUROC | Average Training Time (s) | % of pixel data used in training |
|---|---|---|---|---|
| BAM 1 [2] | $95.70 \pm 2.10$ | – | – | 100 |
| BAM 2 [2] | $97.12 \pm 1.27$ | – | – | 100 |
| All patches | $88.55 \pm 8.60$ | $0.94 \pm 0.06$ | 2110 | 100 |
| Segmented patches | $97.15 \pm 3.24$ | $\mathbf{1.00 \pm 0.01}$ | 2097 | 69 |
| Leaf nodes | $95.65 \pm 2.11$ | $0.99 \pm 0.02$ | **1041** | **32** |
| All nodes | $\mathbf{97.13 \pm 1.21}$ | $0.98 \pm 0.02$ | 1368 | 43 |

Table 2. Performance of CLAM model trained on 4 different sets of extracted patches compared to state-of-the-art from original CRAG paper on two class problem. Haematoxylin channel patches were extracted using the threshold $\mu - \sigma$. Reported metric values are *mean* $\pm$ *standard deviation* on the validation set, the variation is due to the 3-fold cross validation. Training times are averaged across the 3 cross validation folds for 20 epochs with all hyperparameters kept constant.

## 4. Discussion

Figure 3 shows that our qualitative finding of setting the splitting threshold at $\mu - \sigma$ yielded the best results also held true when compared to a wide range of possible thresholds in downstream performance. When the splitting threshold was set too high e.g. $t > (\mu + \sigma)$ we found that frequently the non-cancerous class images would not be split at all resulting in a quadtree with one root node only and a corresponding bag containing a single instance. This occurred because the non-cancerous images had more empty space on average than the cancerous images and contained less regions heavily stained with Haematoxylin. This in turn led to poor performance as the model was operating on a single instance which was heavily downsampled and did not have enough data to learn from and perform well. As this threshold was decreased the images were split more effectively by the algorithm resulting in improved performance. The performance peaked at the qualitatively best performing threshold of $\mu - \sigma$ with the smallest standard deviation between cross validation folds. The performance then decreased beyond this point with a much higher variation across cross validation folds. Inspecting the individual fold performance showed this was driven by one of the three folds performing much poorer than the others; however, this was not always the same fold in each case so this cannot have been occurring due to a difference in the data distribution between folds.

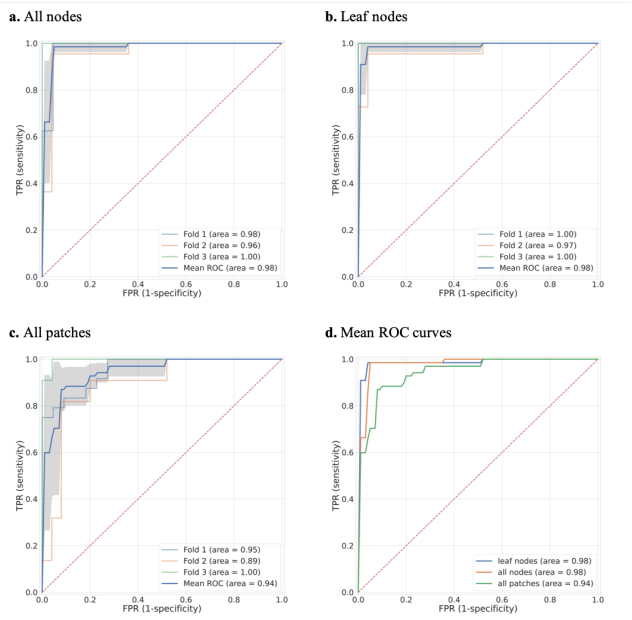


Figure 4. **a, b, c** AUROC curve plots of cross validation average AUROC for CLAM models trained on all quadtree nodes, the leaf nodes from the quadtree and all patches respectively. The quadtree used was the best performing haematoxylin channel method extracted at a threshold of $\mu - \sigma$. Grey shaded area indicates $\pm$ one standard deviation across the three folds. **d** Average AUROC curve for all three methods

We found that when $t < \mu - \sigma$ the algorithm started to split regions that contained a low percentage of tissue where it was not before. This resulted in one patch containing tissue being generated but potentially two or three containing majority or all blank space. We believe this is why performance becomes more varied, more instances are being generated that are not conducive to making an accurate prediction. The threshold of $\mu - \sigma$ appears to be a good middle ground between decomposing the tissue regions accurately and not over decomposing the images. While it is logical that such a middle ground should exist theoretically, we do not yet know why this is found at $\mu - \sigma$. We plan to explore this approach with other data sets in future to see if this finding holds true in different data distributions.

We found that our quadtree image representation is able to achieve better performance than the existing state of the art method for patch extraction in computational pathology. The CLAM model trained with the patches extracted from our quadtree method was able to achieve as good average cross validation performance with a smaller standard deviation compared to a model trained using patches extracted using a tissue segmentation mask while using 38.31% less patches.

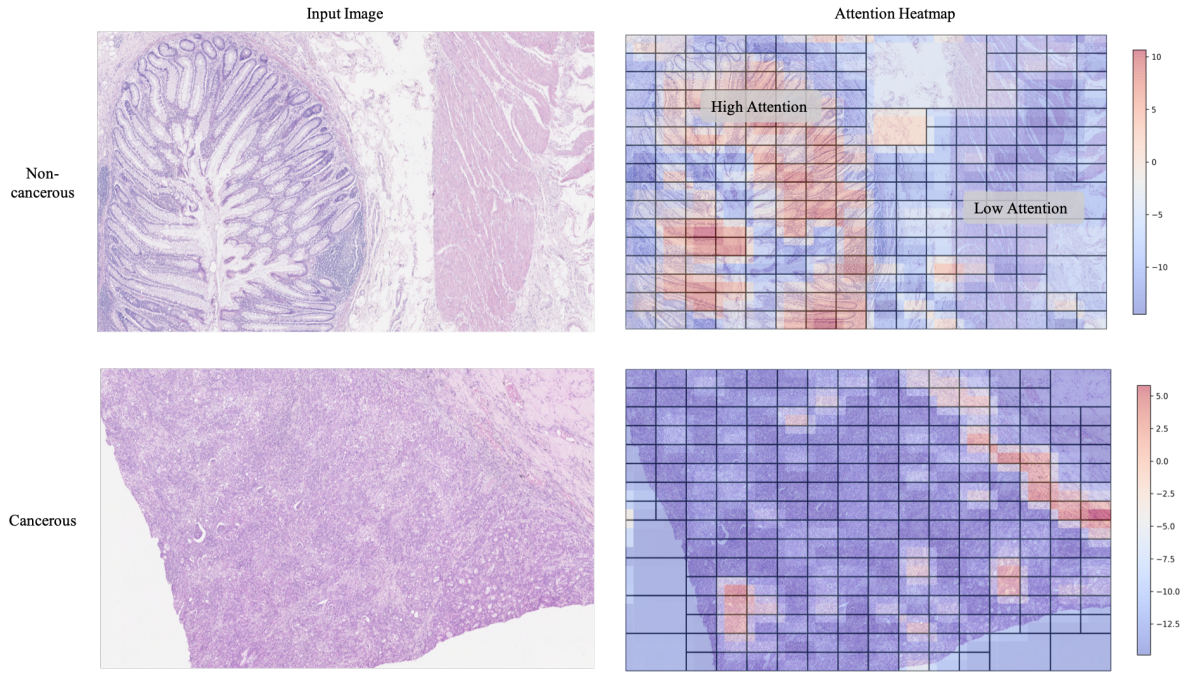


Figure 5. **Attention heatmap visualisations** For a non-cancerous (top) and a cancerous (bottom) image respectively the attention heatmap from the CLAM model trained using the Haematoxylin channel extracted patches at $\mu - \sigma$ overlaid on the quadtree decomposition of the original image (right). We smoothed the attention heatmaps for visibility by using 50% overlap between the patches and averaged the attention weights. The most highly attended regions are denoted in red while the less attended regions are denoted in blue. Regions have been re-scaled back to their original shape and size for display purposes.

When we excluded the non-leaf nodes' patches, the final model performance did degrade slightly but only by 1% compared to when all patches from the quadtree were included. We hypothesise this indicates that the higher level nodes in the tree (at lower magnifications) only contribute marginally to the final prediction. To verify this we overlaid the attention weights from the trained CLAM model in a heatmap over the original images. If the higher level non-leaf nodes were weighted highly we would expect to see large regions of high attention overlaid over regions that have been split further. However, the example in Figure 5 does not show this. It is representative of many other instances in our results where the leaf nodes are clearly the most highly weighted.

The attention heatmap in Figure 5 also shows that the CLAM model trained using our quadtree strategy is able to assign higher attention values to clinically relevant regions. In the non-cancerous example, almost all of the highly weighted patches are located in a region of healthy glands and tissue, highlighted in red and orange, while the stromal region has been coloured a dark blue indicating very low attention has been assigned in this region by the model. This means the model has correctly identified that the presence of healthy glands indicates a non-cancerous region.

## 5. Conclusion

We have shown that quadtrees can be used in computational pathology as an efficient image representation which can be used for fast and highly accurate downstream performance. Our quadtree method is able to decompose histopathology images by identifying regions significant to clinical diagnosis while ignoring less significant regions such as empty space or connective tissue. We show that this image representation is able to be leveraged in a MIL setting to achieve better performance using 38% less data than the currently widely adopted thresholding based tissue mask approach used in the field while also providing an interpretable visualisation of which regions within the image are important for the algorithm in generating its prediction.

In future, we plan to explore this approach with other computational pathology data sets to further verify our findings and explore how the framework performs for other tasks and tissue types as well as how sensitive it is to other staining methods and visual artifacts. If our results here hold true, the method should be very helpful in WSIs to reduce the significant data requirement currently present in processing these large images.

## References


[1] R. Achanta, A. Shaji, Kevin Smith, Aurélien Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:2274–2282, 2012.

[2] R. Awan, K. Sirinukunwattana, D. Epstein, Samuel D. R. Jefferyes, U. Qidwai, Z. Aftab, I. Mujeeb, D. Snead, and N. Rajpoot. Glandular morphometrics for objective grading of colorectal adenocarcinoma histology images. *Scientific Reports*, 7, 2017.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

[4] M. Carbonneau, V. Cheplygina, Eric Granger, and G. Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognit.*, 77:329–353, 2018.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] Thomas G. Dietterich, R. Lathrop, and Tomas Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89:31–71, 1997.

[7] R. Finkel and J. Bentley. Quad trees a data structure for retrieval on composite keys. *Acta Informatica*, 4:1–9, 1974.

[8] S. Graham, D. Epstein, and N. Rajpoot. Dense steerable filter cnns for exploiting rotational symmetry in histology images. *IEEE Transactions on Medical Imaging*, 39:4124–4136, 2020.

[9] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[10] D. J. Ho, D. V. K. Yarlagadda, T. D'Alfonso, M. Hanna, A. Grabenstetter, Peter Ntiamoah, E. Brogi, L. Tan, and Thomas J. Fuchs. Deep multi-magnification networks for multi-class breast cancer image segmentation. *Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society*, 88:101866, 2021.

[11] Maximilian Ilse, Jakub M. Tomczak, and M. Welling. Attention-based deep multiple instance learning. In *ICML*, 2018.

[12] P. Jayaraman, Jianhan Mei, Jianfei Cai, and Jianmin Zheng. Quadtree convolutional neural networks. In *ECCV*, 2018.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[15] G. Litjens, Thijs Kooi, B. E. Bejnordi, A. Setio, F. Ciompi, M. Ghafoorian, J. V. D. Laak, B. Ginneken, and C. Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[16] G. Lu, Dongsheng Wang, X. Qin, S. Muller, James V. Little, Xu Wang, Amy Y. Chen, G. Chen, and B. Fei. Histopathology feature mining and association with hyperspectral imaging for the detection of squamous neoplasia. *Scientific Reports*, 9, 2019.

[17] M. Lu, Drew F. K. Williamson, Tiffany Y Chen, Richard J. Chen, Matteo Barbieri, and Faisal Mahmood. Data efficient and weakly supervised computational pathology on whole slide images. *Nature biomedical engineering*, 2021.

[18] Mustafa Umit Oner, Jared Marc Song Kye-Jet, H. Lee, and Wing-Kin Sung. Studying the effect of mil pooling filters on mil tasks. *ArXiv*, abs/2006.01561, 2020.

[19] N. Otsu. A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9:62–66, 1979.

[20] A. Ruifrok and D. Johnston. Quantification of histochemical staining by color deconvolution. *Analytical and quantitative cytology and histology*, 23 4:291–9, 2001.

[21] B. Schömig-Markiefka, A. Pryalukhin, W. Hulla, A. Bychkov, J. Fukuoka, A. Madabhushi, V. Achter, Lech Nieroda, R. Büttner, A. Quaas, and Y. Tolkach. Quality control stress test for deep learning-based diagnostic model in digital pathology. *Modern Pathology*, pages 1 – 11, 2021.

[22] C. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423, 1948.

[23] S. Tanimoto and T. Pavlidis. A hierarchical data structure for picture processing. *Computer Graphics and Image Processing*, 4:104–119, 1975.

[24] David Tellez, G. Litjens, J. A. van der Laak, and F. Ciompi. Neural image compression for gigapixel histopathology image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:567–578, 2021.

[25] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn. *ACM Transactions on Graphics (TOG)*, 36:1 – 11, 2017.

[26] Peng-Shuai Wang, Chun-Yu Sun, Y. Liu, and Xin Tong. Adaptive o-cnn: A patch-based deep representation of 3d shapes. *arXiv: Computer Vision and Pattern Recognition*, 2018.