university of groningen

faculty of science and engineering

Exploring Vector Attributes of Max tree, Component tree using Supervised Machine Learning Technique

Pooja Gowda

university of groningen / faculty of science and engineering

University of Groningen

Exploring Vector Attributes of Max tree, Component tree using Supervised Machine Learning Technique

Pooja Gowda (S4410963)

March 29, 2023

Master's Thesis

To fulfill the requirements for the degree of Master of Science
in Computer Science at University of Groningen
under the supervision of:
Prof. dr. Kerstin Bunte (Intelligent Systems, University of Groningen)
and
Dr. Michael H.F Wilkinson (Intelligent Systems, University of Groningen)

Contents

Acknowledgments

This work is the result of my parents' blessings, my supervisors' guidance and patience, and the love and support of my friends. Therefore, I dedicate this thesis work to them.

I want to express my sincere gratitude to my esteemed supervisors, *Prof. Dr. Kerstin Bunte* and *Dr. Michael H.F Wilkinson*, for their invaluable guidance and support throughout my research work. Their constructive feedback and patient mentoring have been instrumental in shaping my skills and enhancing my knowledge. I sincerely appreciate their commitment to excellence and their passion for their field of expertise, which has been a constant source of inspiration and motivation for me during my thesis. I am honored to have worked with such eminent personalities and gained immense technical expertise and professional skills under their supervision.

I want to thank my mother, father, and brother for their unwavering trust, love, and unconditional support during my thesis work. Their sacrifices and encouragement have been the foundation of my strength and perseverance. Furthermore, I would like to express my sincere gratitude and affection to my friends, whose unwavering motivation and honesty have helped me navigate through the challenges of this process.

Lastly, I am genuinely grateful for the blessings, time, and support of all those who have been a part of my life, providing me with guidance, encouragement, and positivity.

Abstract

In Europe, lung cancer is prevalent, with men being diagnosed more frequently than women. It is projected to have a mortality rate of 54 per 100,000 people in the European Union in 2020. The mortality rate is correlated with the incidence rate but with a delay caused by low survival rates. This delay is because the absence of lung cancer screening programs hinders early detection and treatment, particularly for those at high risk. In the medical field, Fluorodeoxyglucose Positron Emission Tomography (FDG-PET) scans are primarily employed for detecting malignant, metabolically active lesions. They are also used for staging and monitoring the response to therapy of malignant diseases. A recent study utilized an unsupervised machine learning technique called Self Organizing Maps (SOMs) to classify lung tumors in FDG-PET scans by examining the clusters of vector attributes.

This thesis proposes a general-purpose data science framework using a supervised learning technique, Random Forest, to classify lung tumors and analyze the significance of vector attributes to classify tumors effectively. To analyze the 3D volume data efficiently, component filtering techniques with attribute filtering are applied. Distributed Connected Component Filtering and Analysis (DISSCOFAN) builds the Max Tree data structure for each FDG-PET 3D volume scan data, which are further separated into Training and Test samples as part of preprocessing. The Training samples are used to train the random forest model, and the Test sample is used to evaluate the generalizing ability of the model. The results show that specific nodes of the PET scan volumes are sensitive to morphological features with lung tumors. This sensitivity in identifying lung tumors is more evident when a substantial percentage of nodes with a high tumor fraction, i.e., greater than 10% of total nodes, are utilized for training. Additionally, the sensitivity increases when the threshold for high tumor fraction is set to a tumor volume of 90% or greater relative to the total volume. This indicates that the proposed supervised data science framework can examine vector attributes in max tree data structures to classify tumors effectively without manually thresholding a set of vectors.

Symbols

$E = U$	E is the Universal Set.
$P(E)$	$P(E)$ is family of all binary images(subsets) of E
X	X is a binary image.
ψ	ψ is the connected operator.
$\Gamma_x(X)$	$\Gamma_x(X)$ is binary connectivity openings
$C \subseteq P(E)$	C is set of all connected components subsets of E
λ	λ is attribute threshold
Λ	Λ is attribute measure constraints function of C
Attr(C)	Attr(C) is attribute of connected component C
h	h is threshold level of connected component C
γ^{Λ}	γ^{Λ} attribute opening for grey-scale images
$\boldsymbol{r}$	$\boldsymbol{r}$ reference vector obtained by a reference shape
$\boldsymbol{\tau}$	$\boldsymbol{\tau}$ single vector attribute
ϵ	ϵ is a scalar value
Φ	Φ vector attribute thinning
m_{pqr}	m_{pqr} Raw image moment of order (p+q+r)
$\boldsymbol{A}$	$\boldsymbol{A}$ Array of input vectors
$\boldsymbol{W}$	$\boldsymbol{W}$ Array of weight vectors
n	n output layer of neurons
d	d distance between the input vector $\boldsymbol{A}$ and the weight vector $\boldsymbol{W}$
c	c computed neuron as the winning neuron
b	b topological neighbor neuron from the winning neuron
t	t iteration count value within range of 1 to total iteration
H	H Entropy value
p	p prior probability of each class
g	g number of unique classes
S	s Training sample for random forest algorithm
B	B Total number of trees
e	e iterating index within range of 1 to B
F	F total number of features from the training sample S
u	u Small subset from total features F
o	o set of classifiers
O	O aggregated tree classifier

Chapter 1

Introduction

Lung cancer is the primary cause of mortality associated with cancer globally, representing the most significant proportion of cancer-related deaths at 18.0% [1]. In Europe, approximately 20% of all cancer deaths are attributed to lung cancer [2]. Recent studies have proposed deep learning and image segmentation filtering-based solutions for detecting lung cancer. For example, in [3], the authors proposed a deep learning methodology that uses Convolutional Neural Networks (CNNs) to detect lung tumors in CT scans. By ensembling three CNN models (CNN1, CNN2, and CNN3) and integrating three different deep learning methods employing an ensemble 2D approach, the authors achieved a 95% accuracy in detecting tumors [3]. CNNs are commonly used in artificial intelligence-based solutions to classify image data. They have been applied to identify tumors in pathology images of various cancer types, including head and neck cancer, prostate cancer, and renal cell carcinoma [4]. While in [5], the authors use image segmentation techniques such as thresholding and watershed transform have been used to detect and segment lung nodules from CT scans. On the other hand, In the most recent work in [7], the authors propose a hybrid technique for detecting lung tumors, which combines machine learning with image segmentation techniques. This study introduces an exploratory

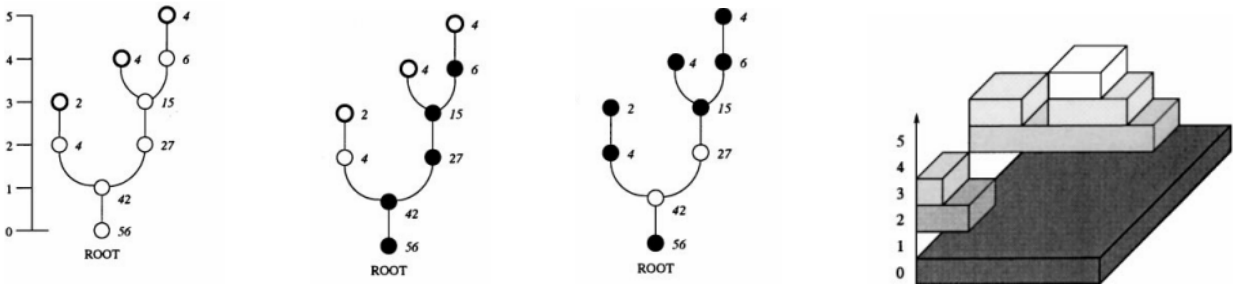


Figure 1.1: Construction and filtering of a component tree: (a) a component tree, where leaves are indicated by the circles in bold at the top; (b) filtering the tree (the filled circles represent active nodes); (c) tree filtered with nonincreasing criterion; (d) perspective view of components corresponding to active nodes [6].

tool for analyzing and exploring the morphological features of the Max Tree and Component Tree, which is built on each Fluorodeoxyglucose Positron Emission Tomography (FDG-PET) scan volume using Distributed Connected Component Filtering and Analysis (DISSCOFAN) [8]. This tool employs an unsupervised machine learning technique, Self Organisation Maps(SOMs), to cluster vector attributes, facilitating the automatic thresholding of vector attribute filtering [7]. The vector Attribute filtering tool is a subset of a connected filter tool [9]. A connected filter is a valuable tool in mathematical morphology that facilitates the selection or elimination of connected components, which are sets of pixels that form a specific region within an image. These tools heavily rely on the notions of flat zone partition and connectivity and are used to filter gray-level images by partitioning them. They are said to serve as a bridge between classical filtering and segmentation, as they are based on partitioning [9]. Component filtering is a valuable image processing technique that involves using an attribute filter to selectively filter images based on size and shape characteristics, as outlined in [10]. This filter is particularly effective at removing undersized or asymmetrical-shaped objects from an image, resulting in a clearer and more refined appearance. Overall, an attribute filter is an essential tool in image processing that can significantly improve image segmentation, filtering accuracy, and overall quality [10].

Many recent works show the application of attribute filtering of an image in several domains, especially in medical areas, to detect organs and cells in an image [11, 12] multi-scale analysis in remote sensing and detecting stellar objects in astronomy. Figure 1.2 briefly shows how attribute filters benefit image segmentation [6]. The background noise is removed as much as possible. Hence, the pixels indicate the face located in the image. Usually, attribute filtering is efficient in a scenario where the discriminating power of attributes is high(i,e., less noisy), and it can be applied effectively to multi-scale representations of images, e.g., Max trees and Component trees [9]. However, in many other cases, multiple images of similar types (low discriminatory power or high noise level) will be put forth for the analysis, which creates difficulty in extracting and comparing the features between the images [10]. Therefore, the set of attributes chosen helps significantly represent the multi-scale data structure (Max Tree, Com-

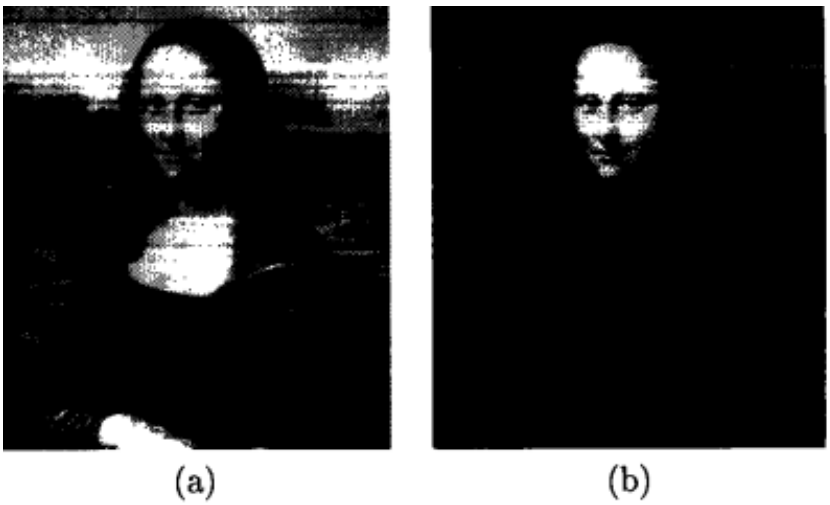


Figure 1.2: Filtering an image using attribute filters [6].

ponent Tree) better, the dissimilarity threshold is set, and attributes describing the given data almost perfectly are selected manually [10]. Therefore, in [7], authors proposed to address this issue and enable the detection of lung tumors effectively using unsupervised machine learning techniques. Building on the previous work by authors in [7], this Master's thesis proposes a data science framework that examines the vector attributes of a Max tree or component tree using the supervised machine learning technique, Random Forest. Furthermore, we investigate the influence of vector attributes on the classifier's performance.

This contribution includes seven major contributions:

1. Exploring and Analysing the Vector Attributes presented in Max tree data structure already available after being built by DISSCOFAN for each PET 3D scan volume.
2. Preprocessing the data samples and creating balanced data samples to train the supervised classifier.
3. Developing the Random Forest model and training the model with different samples using the Majority under-sampling technique.
4. Applying the supervised machine learning technique, random forest, to classify the tumor nodes and analyze the generalizing ability of the classifier on unseen data.
5. Inspecting the vector attribute's influence in classifiers performance.
6. Validating the classifier's performance concerning ground truth volume provided by medical experts.
7. Developing a Proof of Concept to propose a data science framework that helps examine the component tree's vector attributes using a supervised learning technique, Random Forest.

The current report follows a structured approach. Section 2 outlines the methodology employed in the study. Then, in Section 3, the experiments conducted are briefly discussed. The results obtained are presented and analyzed in Section 4. Lastly, Section 5 concludes the report by presenting insightful observations and discussing potential avenues for future work.

Chapter 2

Methods

This section introduces methods and necessary background knowledge to understand the implementation part of the thesis.

2.1 Max Tree Structure

The connected filters do not operate on individual pixels but instead work on connected components of constant gray value in the image [7]. These components are formed by merging flat zones, which are maximal connected regions with a constant gray value. The component tree data structure is used to organize these connected components at different threshold levels of the image, using hierarchical parent affinities between data nodes [9]. The Max tree and Min tree are compact representations of the component tree, with leaves representing regional maxima and regional minima, respectively. Many connected filters, including attribute filters, can be efficiently implemented using these trees [9].

Consider a non-empty universal set E and the family $P(E)$ consisting of all subsets of E, binary images are subsets of $P(E)$ and E [13]. A function f maps a greyscale image I from set E to a subset $T \subseteq \mathbb{R}$, i.e., $f : E \to T$. An operator ψ on a binary image X is called connected if the difference in the set $X/\psi(X)$ only contains connected components of X or its complement X^c. Binary connectivity openings $\Gamma_x(X)$ with a connectivity class $C \subseteq P(E)$ can access these connected components. Here, a *Connectivity Class* C is defined as the set of all connected component subsets of E [13]. Given an intersection point $x \in E$, the connectivity opening $\Gamma_x(X)$ within X can be defined as follows [13]:

$$\Gamma_x(X) = \begin{cases} \cup C_i \in C | x \in C_i \subseteq X, & \text{if } x \in X \\ \emptyset, & \text{otherwise.} \end{cases} \tag{2.1}$$

In the case of binary images, attribute filters are applied to preserve or remove connected components C based on certain attribute measure constraints, such as a specific attribute threshold λ that can be expressed as [13],

$$\Lambda(C) = \text{Attr}(C) \geq \lambda \tag{2.2}$$

where Attr(C) represents the attribute of connected component C. For greyscale images, connected components with a threshold level h are defined as [7],

$$X_h(f) = x \in E | f(x) \geq h \tag{2.3}$$

where f maps the set E to a subset of the real numbers. The attribute opening for grey-scale images, denoted by γ^{Λ}, can be defined based on the binary opening Γ^{Λ} as follows [7],

$$(\gamma_{\lambda}(f))(x) = \sup h \leq f(x) | x \in \Gamma^{\Lambda}(X_h(f)) \tag{2.4}$$

The threshold set has a hierarchy property expressed as [7],

$$X_{h+1}(f) \subseteq X_h(f) \tag{2.5}$$

which enables the representation of images as a tree structure, commonly known as the Max tree [7]. This representation facilitates the efficient storage of images by utilizing the parent-child relationship between two components, A and B, where A is a child of B if it is directly included in B. The single point at which this relationship is established is called the canonical element or level root. Figure 2.1 explains this briefly, where the original image consists of five flat nodes: A, B, C, D, and E, and numbers attached to these nodes represent their corresponding gray level value [9]. The image's maxima are the leaves of the max tree, and concerning the original image, as it has only one maxima, the max tree is represented with just one leaf [9]. The max trees are built using DIStributed Connected COmponent Filtering and ANalysis(DISCCOFAN) [8]. The DISCCOFAN uses both distributed and shared memory techniques. Component trees and attribute filters are computed using these techniques, and we finally obtain the relevant vector attributes [8].

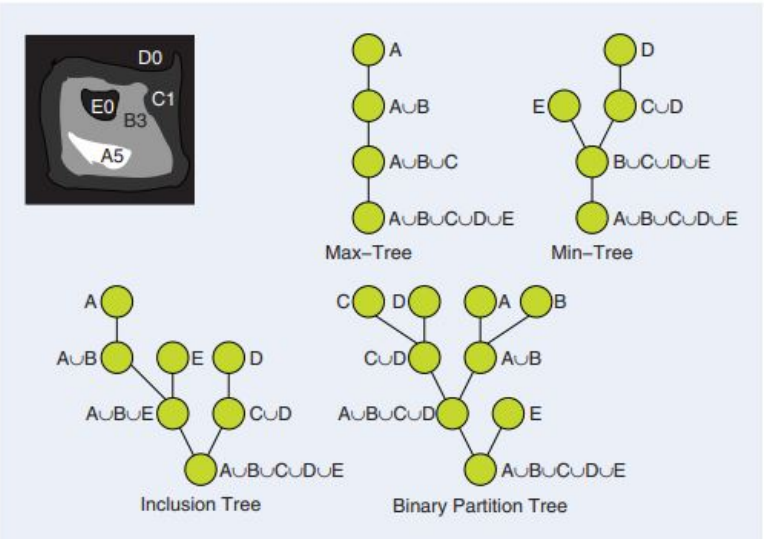


Figure 2.1: Tree representations(Max, Min, inclusion) built from the original images with flat nodes: A, B, C, D, E and the corresponding gray level value [9].

2.2 Vector-Attribute Filters

The author in [10] introduces attribute filters, the subset of the connected filters. These are attributes like the image's shape and size properties, usually will be scalar values. As we already discussed, these filters work efficiently when the images have high discriminating power(less noise) and a more straightforward morphological structure to differentiate [7]. Usually, in many cases, multiple images of similar types or complicated morphological structures will be put forth for analysis, which creates difficulty in comparing the features between the images. Vector Attributes add a more detailed description of the features found in the images, hence improving the discriminatory power [7]. Vector attribute filters, filters based on dissimilarity measures observed concerning vector $\boldsymbol{r}$, are obtained by a reference shape. A single vector attribute $\boldsymbol{\tau}$ of dimensionality D is a set of scalar attributes. While with an attribute threshold vector of dimensionality D, a multivariate thinning is performed. The vector attributes thinning $\Phi^{\tau}_{\boldsymbol{r},\epsilon}$ of X concerning the vector attribute $\boldsymbol{\tau}$ and reference vector $\boldsymbol{r}$ considering the scalar value ϵ is defined as [10],

$$\Phi^{\boldsymbol{\tau}}_{\boldsymbol{r},\epsilon} = \{x \in X, \Lambda^{\boldsymbol{\tau}}_{\boldsymbol{r},\epsilon}(\Gamma_x(X))\} \tag{2.6}$$

The dissimilarity is measured by quantifying the distance between the reference vector $\boldsymbol{r}$ and the vector attribute $\boldsymbol{\tau}$, and the euclidean distance can be calculated [14]. Respectively, the connected components are removed or added depending on the distance of the dissimilarity value computed. We examine different sizes and shape structures in 3D PET scan medical images to improve the discriminating power between images [7]. A number of vector attributes are selected, namely: *Intensity variance*, *Intensity power*, *Intensity mean*, *X-extent*, *Y-extent*, *Z-extent*, *Centre of mass X*, *Centre of mass Y*, and *Centre of mass Z* [7]. The *X, Y, Z-extent* are size-based attributes, which are the differences between the minimum and maximum coordinate values of pixels within each peak component. In the shape aspect, *Flatness*, *Elongation*, *Non-compactness*, *Sparseness*, attributes are considered based on image moments [7], i.e., the center of centroids in X, Y, and Z coordinates. The Raw image moment m_{pqr} of order (p+q+r) of a connected component C for a 3D volume $V(x, y, z)$ of size $L \times M \times N$ is defined as,

$$m_{pqr} = \sum_{C} x^p y^q z^r V(x, y, z) \tag{2.7}$$

Considering the raw moment, The center of mass of the intensity is defined as follows,

$$\{\bar{x}, \bar{y}, \bar{z}\} = \{\frac{M_{100}}{M_{000}}, \frac{M_{010}}{M_{000}}, \frac{M_{001}}{M_{000}}\} \tag{2.8}$$

On the other hand, the central moments on considering the center of mass of the intensity become as,

$$\mu_{pqr} = \sum V(x, y, z)(x - \bar{x})^p (y - \bar{y})^q (z - \bar{z})^r \tag{2.9}$$

The three more variables as statistical dimensions of intensity, namely *Intensity mean, Intensity variance, and Intensity power*. The *Intensity power* of a node N can be calculated as the sum of the squared difference between the intensity value of the node [7].

2.3 Self-organizing Maps - Unsupervised Learning

Kohonen developed a type of neural network called a self-organizing map (SOM) which is used for unsupervised machine learning [15]. The Self-Organizing Map (SOM) can be utilized to map or represent high-dimensional data into a space with lower dimensions. This mapping method maintains the fundamental structure of the original data space, ensuring that comparable data points are positioned near each other in the transformed space [15]. SOM is often used for data visualization, clustering, and dimensionality reduction tasks [16, 17]. Figure 2.2 shows a basic structure of SOM, SOMs have application in various domain which includes sociology [18] and economics [19]. The SOM is made of two layers, where one is an array of input vectors $\boldsymbol{A}$

$$\boldsymbol{A} = [\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_n}]^T \tag{2.10}$$

and another is an output layer of neurons n_k. Here k value ranges from 1,2,3, ... till m. The SOM compares a set of m neurons. An array of weight vectors $\boldsymbol{W_k}$ fully connects the input and output layers [7].

$$W_k = [\boldsymbol{w_{1k}}, \boldsymbol{w_{2k}}, \ldots, \boldsymbol{w_{nk}}]^T \tag{2.11}$$

The SOM learning process follows the following steps [7]:

1. Initialise the weight vectors with random numbers within the range of 0 and 1.
2. Set total iteration counts N.
3. At iteration count value t ranging from j = 1 to N, The distance between the input vector A and the weight vector value W_k is calculated at iteration count t as follows,

$$d_k = \|\boldsymbol{A} - \boldsymbol{W_k}\| = \sqrt{\sum_{j=1}^{m} (\boldsymbol{a_j} - \boldsymbol{w_{jk}})^2} \tag{2.12}$$

4. Compute the output neuron with the shortest distance between the input and weight vector as follows,

$$c(\boldsymbol{a_j}) = \arg\min_j \|\boldsymbol{x_j} - \boldsymbol{w_{jk}}\| \tag{2.13}$$

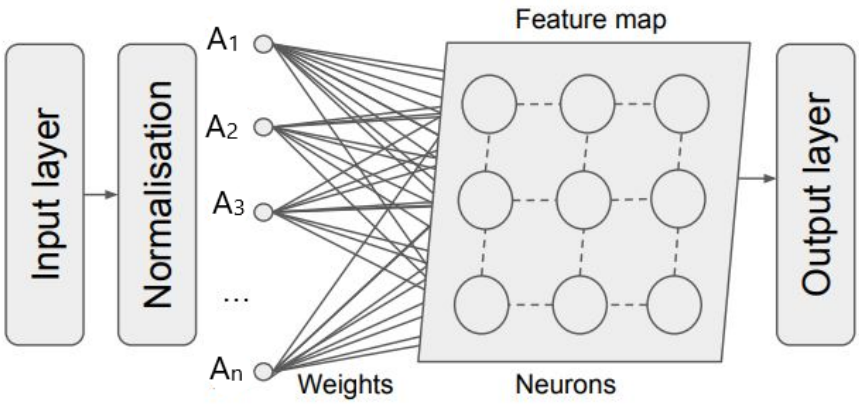


Figure 2.2: Basic Structure of SOM [7].

assign the computed neuron as the winning neuron or the best matching unit (BMU) in the output layer.

5. Compute and update the topological neighbor neurons b_k from the winning neuron. The Gaussian function is usually used for the neighborhood function, and this is used for training the SOMs.

$$b_k = \exp\left(-\frac{\|\boldsymbol{r_k} - \boldsymbol{r_c}\|^2}{2\sigma(t)^2}\right) \tag{2.14}$$

Here the position vectors of neuron n_k and the BMU are $\boldsymbol{r_k}$ and $\boldsymbol{r_c}$. The σ function is the neighborhood spread radius function at iteration count value t.

6. At iteration count value t ranging from j = 1 to N, the weight vector is updated for each neuron as follows,

$$\boldsymbol{W_k}(t+1) = \boldsymbol{W_k}(t) + \alpha(t)b_k(t)(\boldsymbol{A} - \boldsymbol{W_k}(t)) \tag{2.15}$$

The $\alpha(t)$ represents the learning rate at iteration t. The Steps from 3 to 6 are iterated till a maximum number of iterations N [7].

Algorithm 1 Self Organisation Map Algorithm [7].

```
input data vector A;
Initialisation of weight vectors W_k;
N ← Iteration count;
for j = 1 to N do
    d_k ← The distance between the input and weight vector;
    c ← The winning neuron(BMU), that is, the neuron with the shortest distance from the input;
    b_k ← The topological neighborhood of the winning neuron;
    Update the weight vector of each neuron.
end for
```

2.4 Random Forest- Supervised Learning

Tree-based models are an essential component of the random forest algorithm [20]. These models involve recursively partitioning a given data set into two groups based on a specific criterion until a predetermined stopping condition is met. The final nodes of a decision tree are referred to as leaf nodes or leaves. Here, the division process terminates, and the resultant groups are used for prediction, or further analysis [20]. Tree-based models are employed for both classification and regression tasks. The choice of partition and stopping criteria determine the recursive partitioning of a given dataset. The selection of predictor variables to split internal nodes into classification tasks involving categorical outcomes is based on a predefined optimization criterion. Entropy is one commonly used criterion, as explained in [21], the author demonstrates a real-world example of the coding theorem applied to sources, establishing the minimum length required for the binary representation of a random variable. For classification

problems is entropy, which is computed as follows,

$$H = -\sum_{l=0}^{g} p_l \times log(p_l) \tag{2.16}$$

to obtain the highest amount of information at each decision tree split and obtain the best feature, the value of p_l, representing the prior probability of each class, is optimized, given that g denotes the number of unique classes [20]. One of the challenges of using decision trees is that they are prone to overfitting, which occurs when the model captures the nuances and intricacies of the training data, which gives a low classification performance on new, unseen data. This can lead to lower generalization accuracy, which measures how well the model can predict outcomes on data that it has not previously encountered. To enhance generalization accuracy, it is possible to construct multiple individual trees by considering only a subset of observations [20]. The author in [22] first proposed the random-subspace method concept, which the author subsequently elaborated upon in [23] when introducing the formalized version known as the random forest. To arrive at a final prediction, the random forest model aggregates the predictions of multiple individual trees. For optimal performance, it is typically recommended

Algorithm 2 Random Forest Algorithm [24]

Training set S, with F features, and B number of trees in the forest.
function R(a)ndomForest (S, F)
    $O \leftarrow \emptyset$
    for $e \in 1, \ldots, B$ **do**
        $S^{(e)} \leftarrow$ A bootstrap sample from S
        $o_e \leftarrow$ Randomisedtreelearner($S^{(e)}$, F)
        O $\leftarrow$ O $\cup$ o_e
    end for
    return O
end function
function R(a)ndomisedtreelearner($S^{(e)}$, F)
    for each node **do**
        $u \leftarrow$ select a small subset of the feature of F.
        From u, split the best feature
    end for
    return the learned tree
end function

to use several individual trees within the range of 64 to 128 [25]. This range balances achieving a high area under the curve (AUC) metric, minimizing processing time, and keeping memory usage in check [25]. The individual trees are constructed using bootstrap samples rather than the original dataset. This strategy is known as bootstrap aggregating or bagging. In this, the bootstrap technique involves training each tree on a random subset of observations, which is two third of the total sample, known as a bag, instead of using the whole sample [26]. The remaining observations, known as out-of-bag samples, are not used for training that particular tree [20]. Instead, multiple trees are trained on different bags, aggregating their results to

reduce the variance. This aggregation helps to improve the accuracy of the Random Forest model.

The Algorithm 2 briefly shows the algorithm working, Training sample S is considered with F features and B number of trees. For each tree, bootstrap sample S^e is considered, and for each node of the sample, a subset of features u is selected from F. The best feature is selected from the subset of features with higher information gain or reduction entropy computed split. Finally, all the learned trees of different bootstrap samples are aggregated to decide the ensembled set of trees accurately.

2.5 Pattern Spectra

A pattern spectrum summarizes an image's shapes and sizes, which can be computed from granulometric sets of morphological openings or closings [7]. These sets show how the image content changes as the filter's size or shape parameters change and provide information about the distribution of different size and shape classes in the image. The Max-tree method efficiently calculates pattern spectra, where the image is separated into connected components at different threshold levels and stored as an array of nodes. During the tree's construction, attributes can be calculated, and a list of voxels within the image can be saved for visualization [7]. Algorithm 3 demonstrates how to compute pattern spectra using the Max-tree quickly. This information can assign attributes, such as flux, to each node in the 2D pattern spectrum. Figure 2.3 shows an example of a 2D size-shape pattern spectrum created by the switchboard for a diatom image, highlighting the features of the selected bin in the original image [7]. The Max-tree method efficiently calculates pattern spectra by separating the image into connected components at varying thresholds and storing them as an array node, enabling assigning attributes to each node in the 2D pattern spectrum.

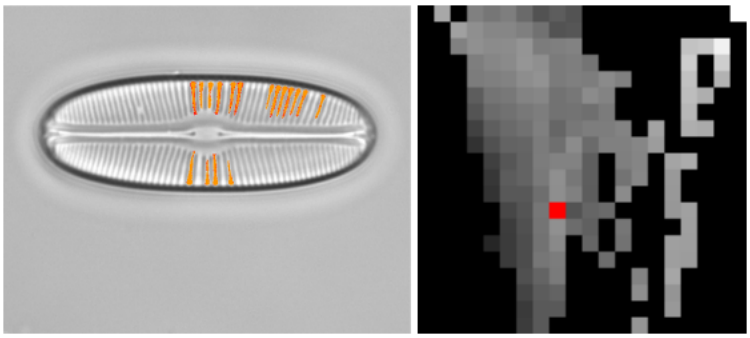


Figure 2.3: Diatom image (left) with its 2D size-shape pattern spectrum (right). In the image, features corresponding to the selected bin in the pattern spectrum are shown in orange [7].

Algorithm 3 The MT_pattern_spectrum_2D [7]

```
1: Precondition: The max-tree is stored as an array of nodes node. Each node contains an
   array of attributes attrib with a length of at least 2.
2: BinFuncAttr1 and BinFuncAttr2 compute the bins in the spectrum to which a node
   should be assigned.
3: function M(T)_pattern_spectrum_2D( MTnode *node, greyval **Spectrum)
4:     Set all elements of Spectrum to zero;
5:     for all node[v] except root do
6:         par = node[v].parent;
7:         flux = (node[v].Gval - node[par].Gval)*node[i].area;
8:         binX = BinFuncAttr1(node[v].attrib[0]);
9:         binY = BinFuncAttr2(node[v].attrib[1]);
10:        Spectrum[binX][binY] = Spectrum[binX][binY] + flux;
11:    end for
12: end function
```

2.6 3D volume visualization Tool: MTdemo

In this thesis, we visualize the classified and ground truth 3D volumes using MTdemo 1.4.0. The authors provide this tool as an open source for research purposes [27].Various visualization modes are available for displaying images, including Maximum Intensity Projection (MIP), Isosurface, and X-ray rendering. In addition, these visualization techniques are used in conjunction with attribute filtering on Max Trees [27]. In this thesis, we have mostly visualized the 3D volumes using the X-ray visualization mode, as shown in Figure 2.4. X-ray rendering is a method of simulating the process of capturing X-rays of a volume data set. Each pixel on display represents a ray that passes through the volume, and the values are added along the ray's path. After normalization, the result is passed through a look-up table (LUT) and displayed on the screen. The default setting uses the color LUT, but this can be disabled. The depth cueing feature can make objects in the distance appear darker than those closer, giving a sense of depth. The intensity mapping controls can adjust the brightness of faint features by adjusting a gamma-like parameter [27]. The parameter can be adjusted using a slider or an input field.

2.7 Previous Related Work

In prior work in [7], the authors introduced a versatile data analysis tool that leverages self-organizing maps (SOMs) in unsupervised machine learning to investigate clusters within the vector attributes of a Component tree, Max tree. Vector attribute filters were used to enhance the performance of traditional attribute filters based on scalar attributes to explore the possibility of automatic detection of tumors in FDG-PET 3D volumes using an unsupervised learning technique [7]. They presented their results in their work's chapter *Experiment and Discussion*. The outcomes demonstrated that trained SOMs with data samples mentioned in table 2.1 were sensitive to specific features in FDG-PET scans, such as organ intensity or nodes with higher

Table 2.1: Four different samples were used for training SOMs [7].

Sample	Total nodes	High tumour nodes	High tumour percentage
1	50,00,000	10,285	0.2%
2	1977900	197790	10.0%
3	988950	197790	20.0%
4	395580	197790	50.0%

Intensity Mean value. Furthermore, when the self-organizing map (SOM) is trained with a large proportion of nodes that correspond to high-fraction tumors (more than 10% of all nodes) in the training set, certain neurons tend to exhibit an increased response to the presence of lung tumors, according to the findings presented in the chapter on the previous work [28]. In this work, authors preprocess these FDG-PET 3D volume scans by building a Max tree data structure for each PET scan using DISCCOFAN [8]. In addition, the relevant vector attributes discussed in section 2.2 are computed from the built Max tree data structure. The chapter provides a comprehensive account of the experiments, results, and authors' discussion of their findings [28].

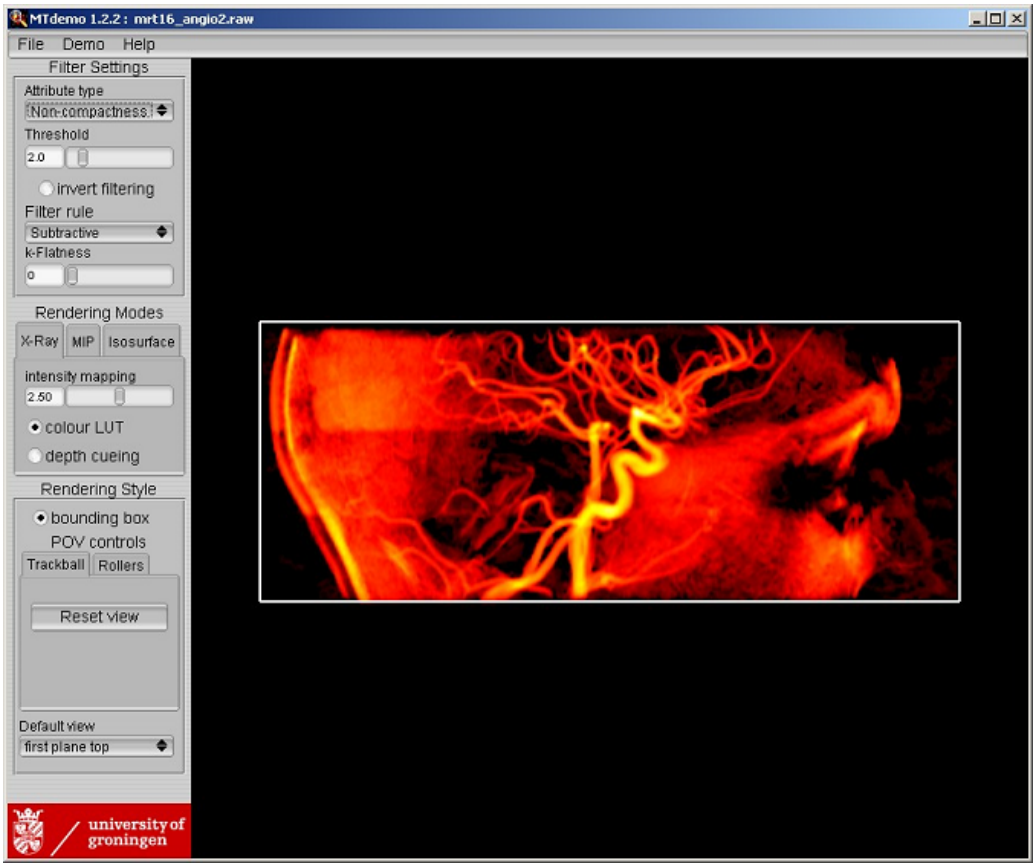


Figure 2.4: X-ray visualization mode using MTdemo visualization tool [27].

2.7.1 Data set

The fluorodeoxyglucose positron emission tomography (FDG-PET) scan is used for this thesis and in previous work by authors in [7]. FDG-PET scans detect lung tumors since FDG-PET can provide information on active inflammatory lesions. Lesions include ovarian, colorectal, lung, melanoma, breast, lymphoma, and brain cancer [29]. The analysis was based on 202 FDG-PET 3D volume scans with two corresponding ground truths (lung tumor images from PET scans by an expert).The ACRIN Cooperative Group (now part of ECOG-ACRIN) and RTOG Cooperative Group (now part of NRG) conducted a multi-center clinical trial that collected FDG-PET scans for pre- and post-chemoradiotherapy imaging [30, 31]. The study's primary objective was directly investigating the clinical outcomes following definitive chemoradiotherapy. The collected FDG-PET scans were from patients aged 18 years and older with non-small cell lung carcinoma following American Joint Committee on Cancer (AJCC) criteria stage IIB/III and scheduled for definitive concurrent chemoradiotherapy due to inoperable disease [7]. Figure 2.5 shows the visualization of one of the FDG-PET scan volumes 197 from different angles.

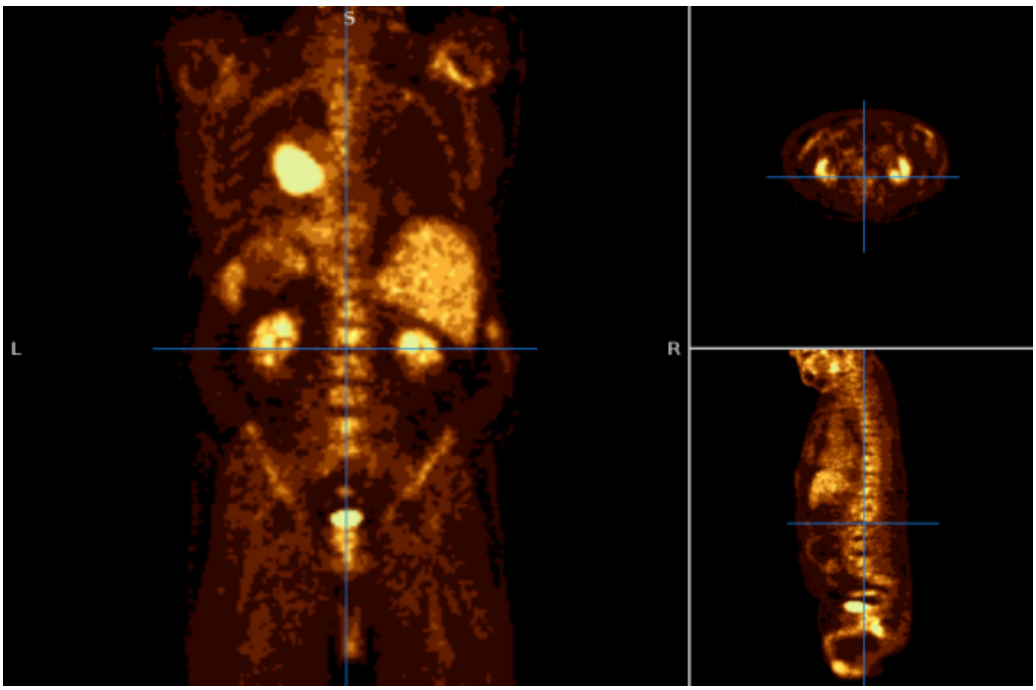


Figure 2.5: 3d PET scan of volume 197 in three different angles (from left, top right, and top-bottom, corresponding to the coronal, axial, and sagittal view) [7].

Chapter 3

Experiments

This section discusses briefly the experiments that were implemented in the thesis to answer the research questions.

3.1 Preprocessing

Throughout this thesis, we use the same datasets used in [7] and regarding, which we briefly explained in section 2.7.1. In addition, we use the Max tree data structure built on each PET scan with computed vector attributes as mentioned in section 2.2 from the Max tree. In the PET scan volumes, it is observed that the lung tumor nodes occupied a significantly smaller portion of the PET scans ($< 5\%$). Similar to the author in [7], in this work, the Nodes with High tumor Fraction (a node with a tumor volume $\geq$ to 90% of the total volume) are considered, which accounts for only around $\sim 0.21\%$ of the total input nodes. To consider the nodes with a Tumour fraction of 90% and above, each volume data set is preprocessed by labeling the nodes $\geq 90\%$ Tumour Fraction as *High*, and the rest are labeled as *Low*. The data set is labeled respectively by adding a feature to each data volume named *Tumour Percent*, where concerning each node's Tumour Fraction indicates whether it is a High tumor or not. In this thesis, the same data sampling method, which is the majority undersampling technique, is used as of [7] as you can observe in Table 2.1, but for our experiment, the data set of 202 volumes is split into 180 datasets as a training set, and the rest 22 volume data set is kept aside as the test set. This helps to evaluate the generalizing ability of the supervised machine learning classifier on an unseen data set. Table 3.1 shows the samples for training the random forest model.

Table 3.1: Four different samples were used for training Random Forest Model.

Sample	**Total nodes**	**High tumor nodes**	**High tumor percentage**
1	5000000	10,285	0.2%
2	1783550	178355	10.0%
3	891775	178355	20.0%
4	356710	178355	50.0%

Table 3.2: Six different samples labeled with different Tumor Fraction Threshold that were used for training the Random Forest Model

Sample	Total nodes	High tumor nodes	Tumour Fraction Threshold	High tumor percentage
1	866862	433431	$\geq 10\%$	50.0%
2	670512	335256	$\geq 30\%$	50.0%
3	540792	270396	$\geq 50\%$	50.0%
4	435136	217568	$\geq 70\%$	50.0%
5	396816	198408	$\geq 80\%$	50.0%
6	356710	178355	$\geq 90\%$	50.0%

3.2 Training Random Forest Model with Biased Data Samples

In this experiment we used training set which consists of 180 data volumes for the training the model and rest 22 data volumes serves as hold out test set. In this training set, High tumor nodes accounts for a smaller portion of datasets, since we consider the nodes with tumour volume $\geq 90\%$ of total volume. This creates an imbalance in the dataset. As part of the experiment and to assess the model's performance on four biased samples with High tumor fraction nodes, i.e., nodes accounting for tumor fraction $\geq 90\%$ as shown in Table 3.1. In the first sample, 50,00,000 nodes were randomly selected from the training set, of which 10285 nodes were classified as High tumor nodes, making up 0.2% of the total High tumor percentage. In the second to fourth sample, the total High tumor nodes, which were 178355 in the training dataset, were considered. Then, nodes were randomly selected from the remaining nodes labeled as low tumor nodes in the training dataset, accounting for 10%, 20%, and 50% of the samples, respectively. The sampling technique used here is Majority Undersample Technique. This technique was also adapted in previous work by the authors in [7] and delivered promising results, which became the basis of applying this technique in supervised learning. The Random forest model is trained with different samples that we discussed and bootstrap samples are chosen from these data samples. The model uses 100 trees to train the data set and ranks the feature subsets. The ranked features help classify the tumor nodes as high or Low (1 or 0).

While training dataset samples that we mentioned previously. We apply 10-fold cross-validation to validate hyper parameters of the model. The total training sample with different High tumor percentages is split into ten folds. These folds are used as validation sets to validate the model and evaluate the overfitting issues. Each fold is used as a validation set at one point. As we already know, the random forest algorithm is such that it chooses a subset of features from total features for every tree learner while training the dataset with features and ranks the features accordingly. These ranked features or predictor's importance enables the algorithm to classify the trained data to classify accurately. As per the feature's relevance, we filter out the least important features to evaluate the vector attribute's relevance of other features which ranked higher to deduce a high classifier performance respectively. After filtering the least essential features, we retrain the model with features ranked higher by the classifier to analyze and estimate the relevance of features accounting for better classification performance. As mentioned already, 22 data volumes serves as hold out test set to evaluate the classifier's generalizing ability on unseen data the trained model has never encountered. In this experiment,

we combined all 20 sets into one whole test set and assessed the classifier's performance on the test set.

3.3 Training Random Forest Model with datasets of different thresholds

Previously, we trained the model on the dataset volumes, which consisted of nodes with tumor volume accounting for $\geq 90\%$ of total tumor volume. Accordingly, the nodes were labeled as *High* and the rest as *Low*. As part of experimentation and to assess the effect of threshold choice of tumor fraction on the classifier's performance on unseen data volumes, we change the threshold to $\geq 10\%$, $\geq 30\%$, $\geq 50\%$,$\geq 70\%$, $\geq 80\%$ and train the model separately with 180 datasets labeled concerning different thresholds, i.e., with six different training samples as shown in Table 3.2 and analyze the classifier behavior on the 22 volumes data as a whole and using k-fold cross validation during training. In this experiment, we train the model similarly as mentioned in section 3.2, but here we biased the data sample such that we considered the total High tumor nodes labeled as *High* in 180 volume data sets and randomly sampled nodes from the rest of the total nodes labeled as *Low* from training set account for 50% of the sample. These biased data samples of every threshold change are used to train the classifier respectively. Also, the total number of High tumor nodes changes due to a change in the threshold we set to label the nodes as *High* and *Low* during the preprocessing stage.

3.4 Performance Validation using Ground Truth PET scan volumes

After evaluating the classifier's performance on the test set and during training using 10-fold cross-validation, we validate the performance by comparing classified volumes from the test set and the training set concerning two ground truth volumes, i.e., Lung tumor images from PET scans provided by the two experts, respectively. We calculate True Positive, True Negative, False Positive, False Negative, F-score, Precision, Recall, and Accuracy. Finally, The average F-score and classification accuracy on the test and training set are computed. These computations are performed concerning every classifier we have trained on datasets with 180 datasets labeled concerning different thresholds, as mentioned in section 3.3. We first filter out the volumes and assign tumor labels(0 or non-zero) according to the classification results to each volume. Then we read the volumes, i.e., filtered classified volume and ground truth volume provided by the expert, and we finally compare the volumes, respectively. Since we have two ground truth volumes for each volume, we compare the classified volume with both the ground truth volumes and compute the True Positive, True Negative, False Positive, False Negative, F-score, Precision, Recall, and Accuracy, respectively.

Chapter 4

Discussion & Results

In this section, we will briefly discuss the results that are obtained from the experiments mentioned in section 3.

4.1 Classifiers trained with biased data samples

In Figure 4.1, the SubFigures 4.1a to 4.4f illustrate the results of the classifiers on the test set, which were trained with biased data samples 1 to 4. From subFigure 4.1a to 4.4f, the classification results on the unseen data clearly show a gradual decrease in the false positive rate. Although very interestingly, in subFigure 4.1a, the false negative rate is meager, the false positive rate is pretty high, i.e., around 155307 nodes are predicted as false positives. Even though the false negative rate of the classification increases when the classifier is trained with biased data sample 2, it gradually decreases to 1509 false negative nodes in the classification result for the classifier trained with data sample 4, as shown in subFigure 4.4f. As we know to determine a classifier's performance, we look for the classifier with a low false negative rate along with substantially low false positive rates as well, on analyzing the classification results of the classifiers that have been trained with different biased samples, it was evident that the classifier trained with 50 percent high tumor data sample performed overall better than other classifiers trained with other biased data samples. Figure 4.3 demonstrates the false positive rate and false negative rate trend with respect to different classifiers results on the test set, which is trained with the four data samples as mentioned in Table 3.1 accounting for different High tumor percentage.

Whereas Figure 4.2 shows how the classifier trained using a random forest algorithm with different biased samples 1 to 4 ranks the features for classifying tumor labels to *High (1)* or to *Low (0)*. The subFigure 4.2c shows features' importance when the classifier is trained with 0.2 percent of high tumor nodes, i.e., with sample 1, here the vector attributes *Elongation of the tumor, Intensity Mean, and Centre of Mass Z* have a higher feature ranking comparative to rest. Meanwhile, the features *Intensity Variance* and *Center of Mass Y* are even significant vector attributes for classification. But when observed for classifiers trained with samples from 2 to 4, the vector attribute *Intensity Mean* appeared to be ranked highest for classification, and gradually, the significance of *Centre of Mass Z* vector attribute or feature appears to reduce.

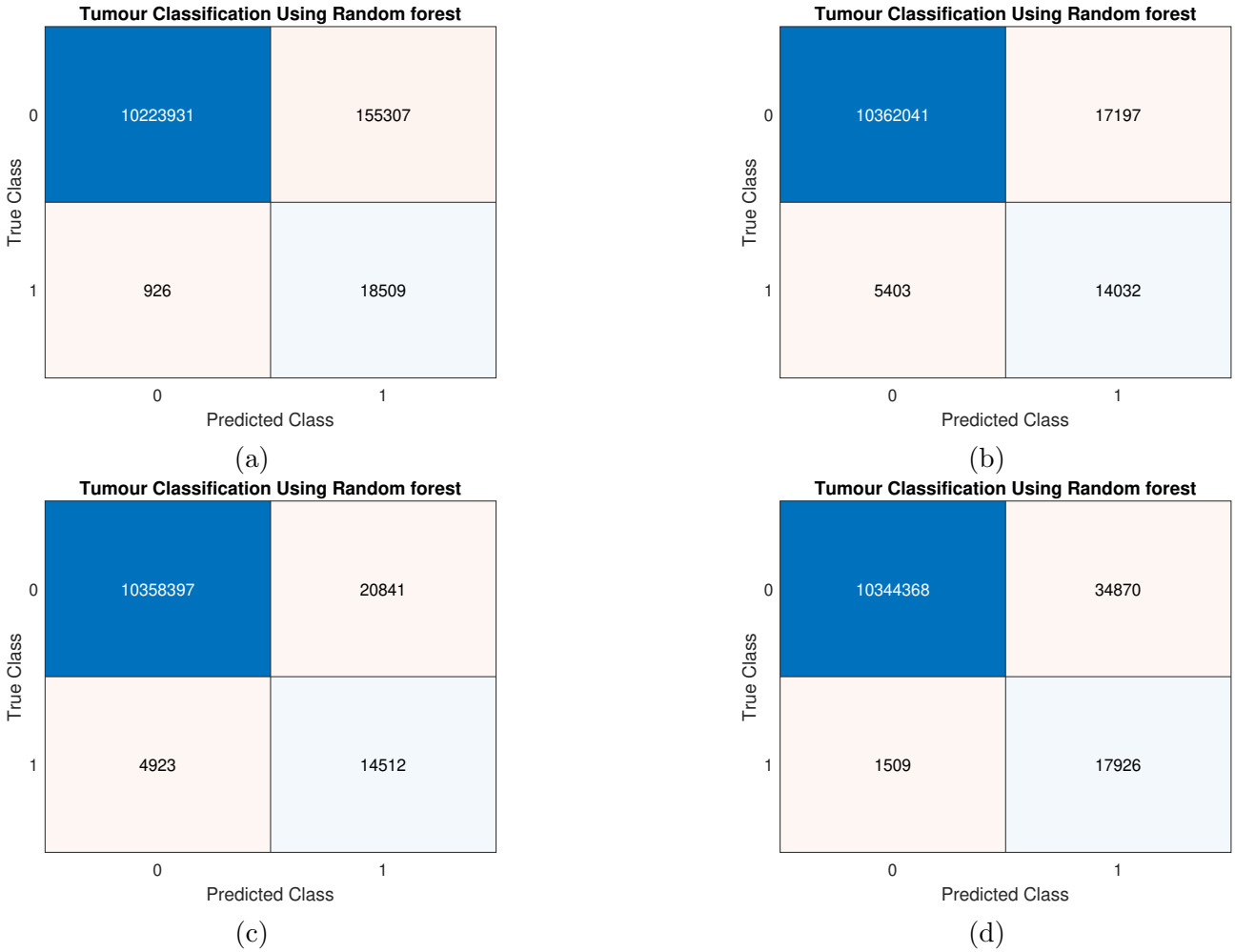


Figure 4.1: The Sub Figures 4.1a, 4.1b, 4.1c, 4.1d are the classification results on the test set, the classifier trained on biased data samples 1 to 4 with tumor volume accounting $\geq 90\%$ of total volume respectively.

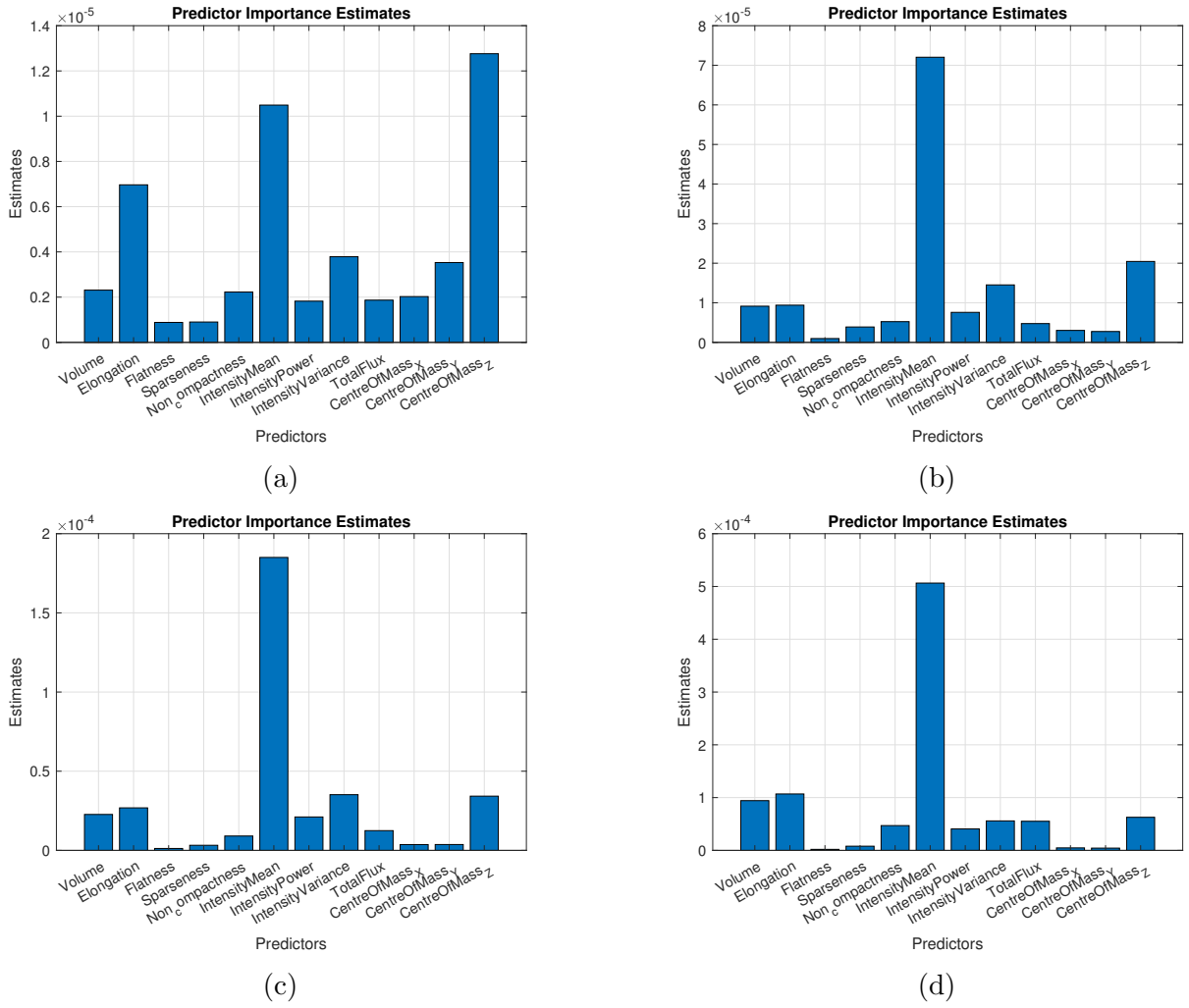


Figure 4.2: The Sub Figures 4.2a, 4.2b, 4.2c, 4.2d show the feature vector ranking by the Random Forest classifier, which is trained with biased data samples 1 to 4 with tumor volume accounting $\geq 90\%$ of total volume respectively.

4.2 Classifier trained with datasets of different high tumor fraction thresholds

In Figure 4.4, we observe the classification results on the test set as shown in the sub Figures 4.4a, 4.4b, 4.4c, 4.4d, 4.4d, 4.4e, 4.4f of the classifier that is trained on 6 data samples as shown in Table 3.2 respectively. As per the results, the false negative rate is higher in the classification results of the classifier trained on sample 1 with tumor volume $\geq 10\%$ of the total volume, and it gradually decreases when the classifier is trained with labeled sample 6 where high tumor fraction is set to tumor volume $\geq 90\%$ of the total volume. Conversely, the false positive rates remain low comparatively until the threshold to be set at tumor volume accounts for $\geq 30\%$ from the total volume, and gradually there is a slight increase in false positive as the threshold gets stricter, i.e., from $\geq 50\%$ to $\geq 90\%$ of tumor volume in total volume. Using Figure 4.6, we show the trends of the false positive rate(%) and false negative rate (%) with respect to different classifiers results on the test set, which is trained with the six data samples as mentioned in Table 3.2, accounting for different High tumor fraction thresholds.

Whereas Figure 4.5 shows how the classifier trained using a random forest algorithm with different samples 1 to 6 of different threshold ranks the features for classifying tumor labels to *High (1)* or to *Low (0)*. The subFigure 4.5a shows features' importance when the classifier is

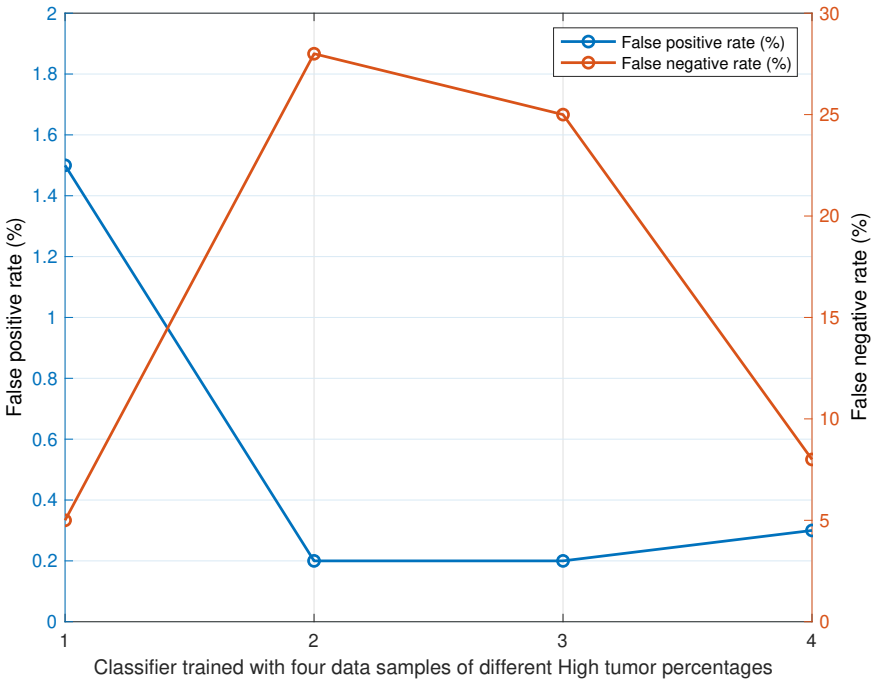


Figure 4.3: False Positive Rate(%) and False Negative Rate(%) results on a test set of the classifiers trained with four data samples accounting for different High tumor percentages. Here, the X-axis depicts the classifiers trained on four different samples, and the Y-axes depict the respective False Positive Rates(%) and False Negative Rates(%).

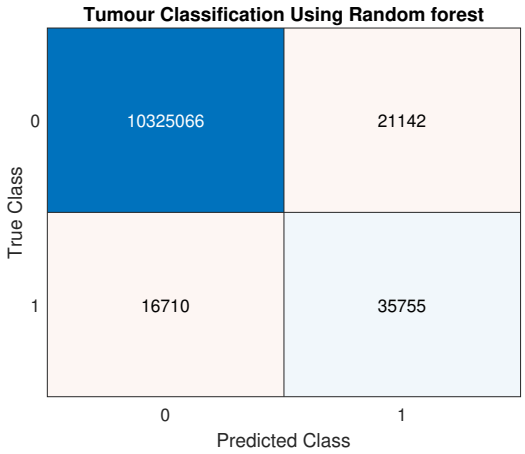
trained with the labeled sample 1 from Table 3.2 where high tumor fraction is set to tumor volume $\geq$ 10% of the total volume, here the vector attributes *Intensity Mean, and Centre of Mass Z* have a higher feature ranking comparative to rest of vector attributes. But, the vector attribute *Centre of Mass Z* significance in classification suddenly decreases as shown in sub Figure 4.5f when the classifier is trained with the labeled sample 6 from Table 3.2 where high tumor fraction is set to tumor volume $\geq$ 90% of the total volume. This also states that when the tumor fraction is lower, i.e., when we consider training the classifier with a labeled sample with a high tumor fraction to be tumor volume $\geq$ 10%, the classifier considers the *Centre of Mass z* vector attribute more relevant for classifying the high tumor nodes when compared to a labeled sample with a high tumor fraction to be tumor volume $\geq$ 90%, that might be precisely due to the location where lungs can be present in the FDG-PET 3D volume, so several nodes located in the area of lungs are also considered since threshold that we have set is too low.

4.3 Performance Evaluation using Ground Truth PET scan volumes

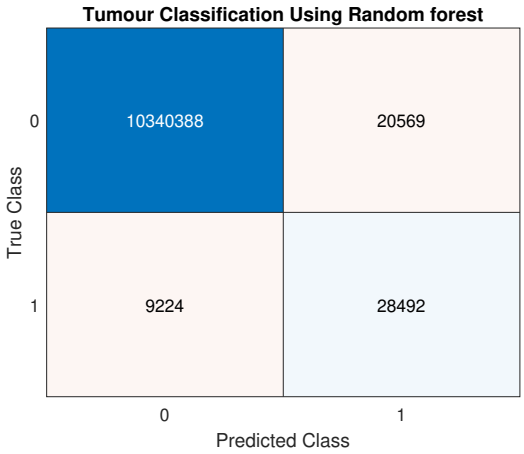
The subFigures 4.7a and 4.7b show the classifier's performance which is trained with different samples as shown in the Table 3.2 validation results on the training set since we split the training set to 10 different folds for cross-validation while training the classifier and on the test set or unseen data volumes. Comparing the classified test set volumes from the classifier, which is trained with the labeled sample 6,5,4 from Table 3.2 with the respective ground truth provided by expert one and expert 2. The average F-scores computed were higher, around 0.42 to 0.45, than that of classifiers trained with other samples. Also, the same trend is observed when comparing the classified training set volumes with the average F-score obtained. On the other hand, the average F-score computed on training and as well as on the test set reduces drastically when the classifiers are trained on samples with *Tumour Volume Percentage of Total Volume or Tumour Fraction Threshold* is from $\geq$ 50% to $\geq$ 10%.

Whereas, Figure 4.8, 4.9 shows the performance of the classifier trained with sample 6 from Table 3.2 with tumor fraction threshold $\geq$ 90% on test set volumes 228 and 238. In brief, the subFigure 4.8c, 4.9c shows the filtered volumes 228 and 238 with the respective classification predictions, which are visualized using MTdemo. On the other hand, subFigures 4.8a and 4.9a are respective ground truth volumes by expert 1, and subFigures 4.8b and 4.9b are respective ground truth volumes by expert 2. Comparing these volumes with ground truth volumes resulted in high fscores. For 228, the fscores obtained were 0.85 and 0.88 with an accuracy of 99%, and for 238, the fscores obtained were 0.74 and 0.76 with an accuracy of 99% for both ground truth volumes. From the subFigure 4.8c, the brighter yellow part of the structure shows the higher number of true tumor votes, i.e., True positive nodes. As the structure becomes reddish or less bright, it indicates fewer true positive nodes. In other words, a brighter structure shows more evidence of the presence of a tumor.

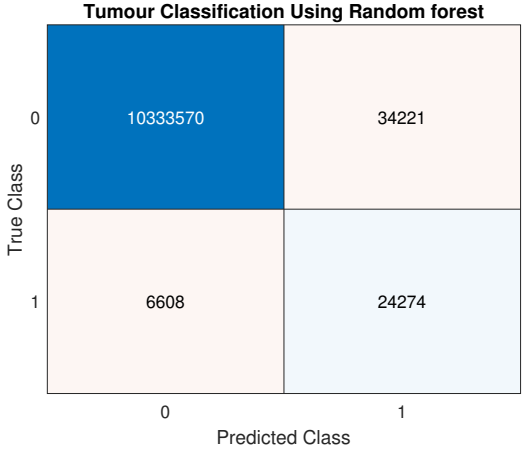
The average fscore computed in subFigure 4.7b is lower because the set of test volumes has a higher false positive rate. This higher false positive rate is observed in all classifiers trained with different samples of different thresholds, which causes the average f-score to decrease gradually. Therefore, we focus on the classifier, which results in to least false negative rate, which is



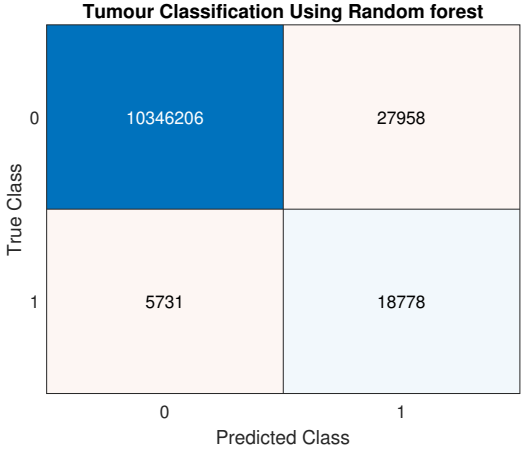
(a)



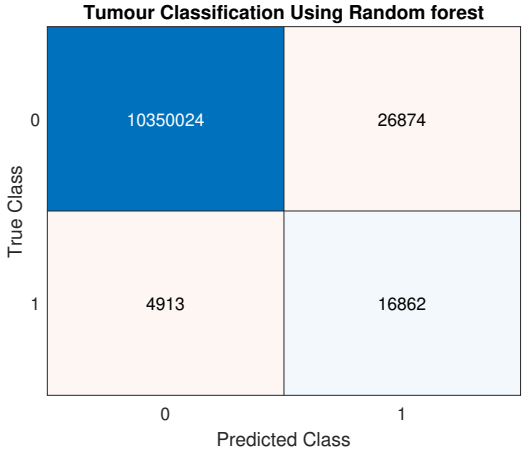
(b)



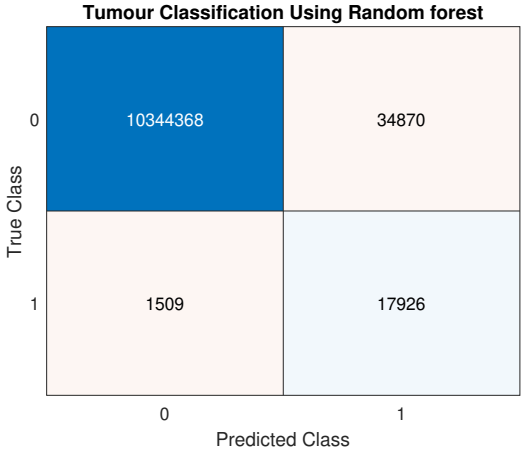
(c)



(d)

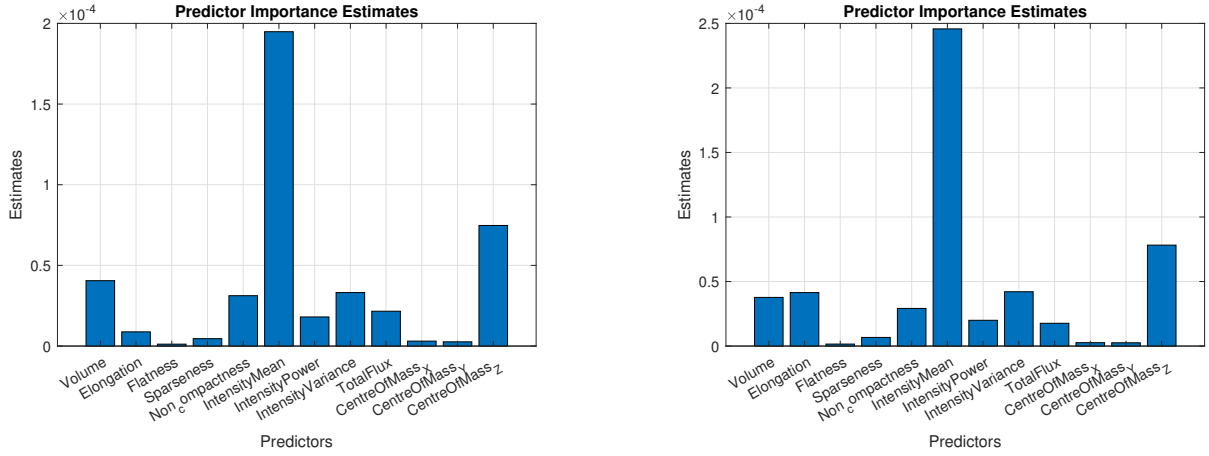


(e)



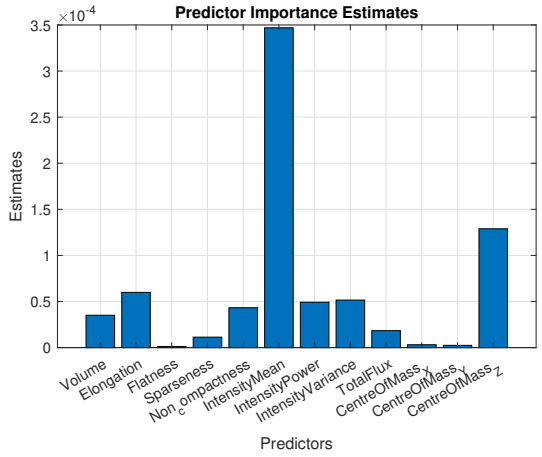
(f)

Figure 4.4: The Sub Figures 4.4a, 4.4b, 4.4c, 4.4d, 4.4e, 4.4f are the classification results of the classifiers on the test set which is trained with the training set with tumor volume accounting for $\geq 10\%$, $\geq 30\%$, $\geq 50\%$, $\geq 70\%$, $\geq 80\%$, $\geq 90\%$ of total volume respectively.

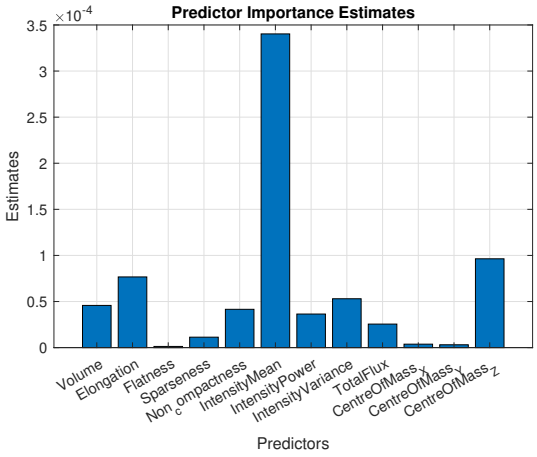


(a)

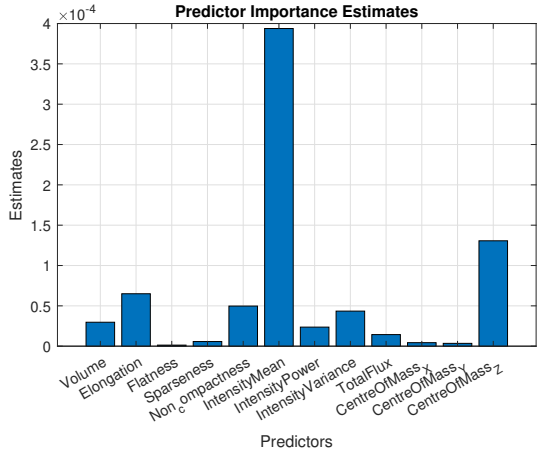
(b)



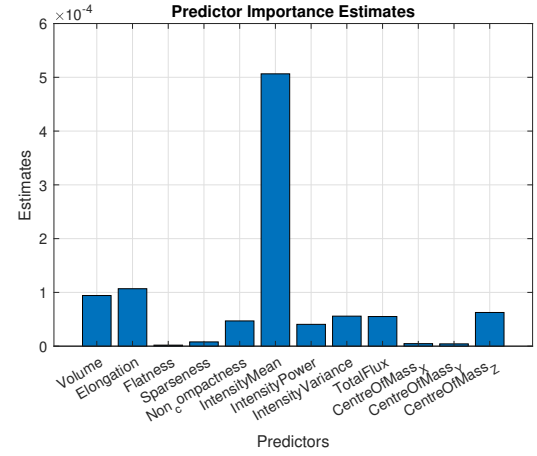
(c)



(d)



(e)



(f)

Figure 4.5: The Sub Figures 4.5a, 4.5b, 4.5c, 4.5d, 4.5e, 4.5f show the feature vector ranking by the Random Forest classifier which is trained with the training samples 1 to 6 with tumor volume accounting for $\geq 10\%$, $\geq 30\%$, $\geq 50\%$, $\geq 70\%$, $\geq 80\%$, $\geq 90\%$ of total volume respectively.

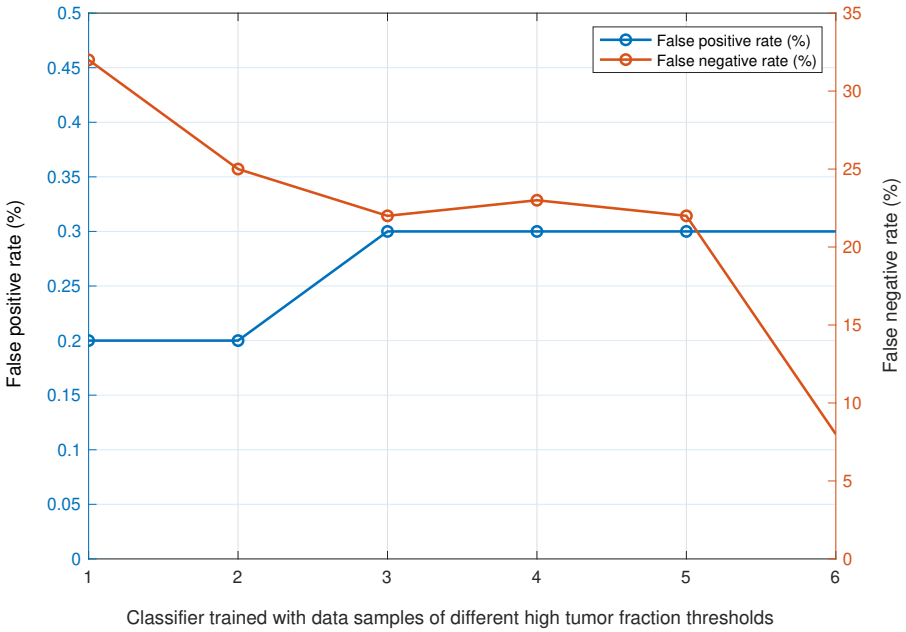
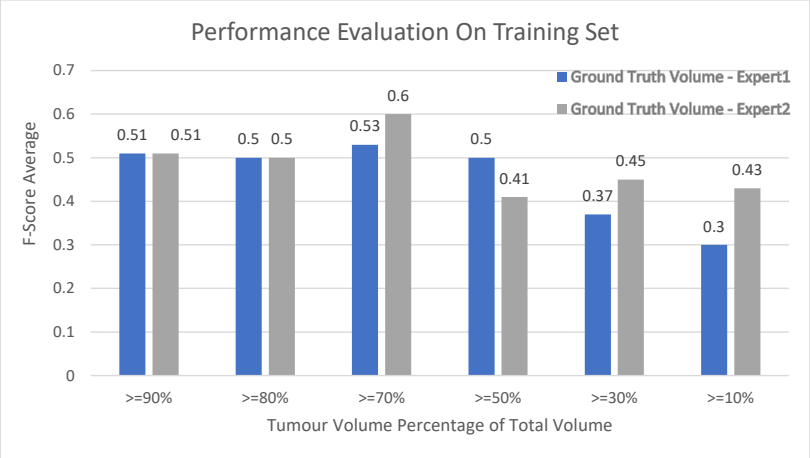
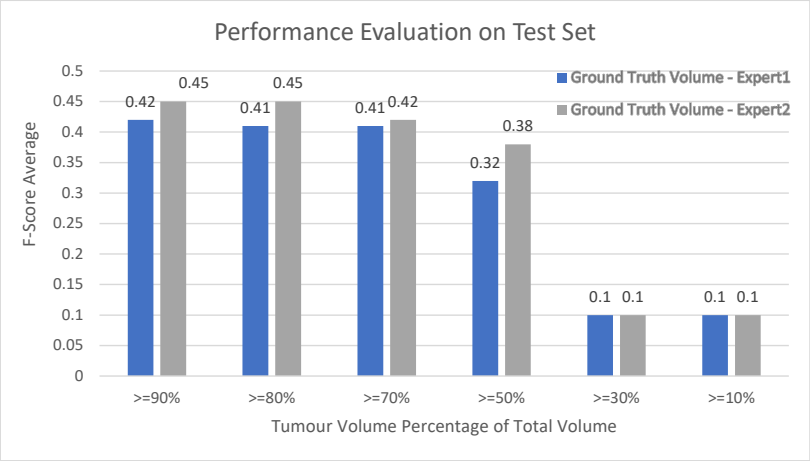


Figure 4.6: False Positive Rate(%) and False Negative Rate(%) results on a test set of the classifiers trained with six data samples accounting for different High tumor fraction thresholds. Here, the X-axis depicts the classifiers trained on six different samples, and the Y-axes depict the respective False Positive Rates(%) and False Negative Rates(%).

the classifier trained with data sample 6. We use this classifier to evaluate its performance against the ground truth volumes provided by expert 1 and expert 2. Figure 4.10, 4.11, and 4.12 shows the performance of the classifier trained with sample 6 from Table 3.2 with Tumour Fraction Threshold $\geq$ 90% on test set volumes 236, 230, 244. In brief, the subFigure 4.10c, 4.11c, and 4.12c shows the filtered volumes 236, 230, and 238 with the respective classification predictions, which are visualized using MTdemo. On the other hand, subFigures 4.10a,4.11a, and 4.12a are respective ground truth volumes by expert 1, and subFigures 4.10b, 4.11b and 4.12b are respective ground truth volumes by expert 2. On comparing to ground truth volumes, these volumes resulted in low fscores. For 236 volume, the fscores obtained were 0.3 and 0.4 with an accuracy of 99%, for 230 volume, the fscores obtained were 0.3 and 0.2 with an accuracy of 99%. Finally, on 244 volume, the fscores obtained were 0.4 and 0.4 with an accuracy of 99%. The false positives indicated the classifier's sensitivity to the nodes with high *Intensity Mean* value. This behavior is consistent even while comparing the training set volumes with ground truth volumes. The visualizations of these volumes and a few other test set volumes with respect to their ground truth volumes are present in the Appendix section.

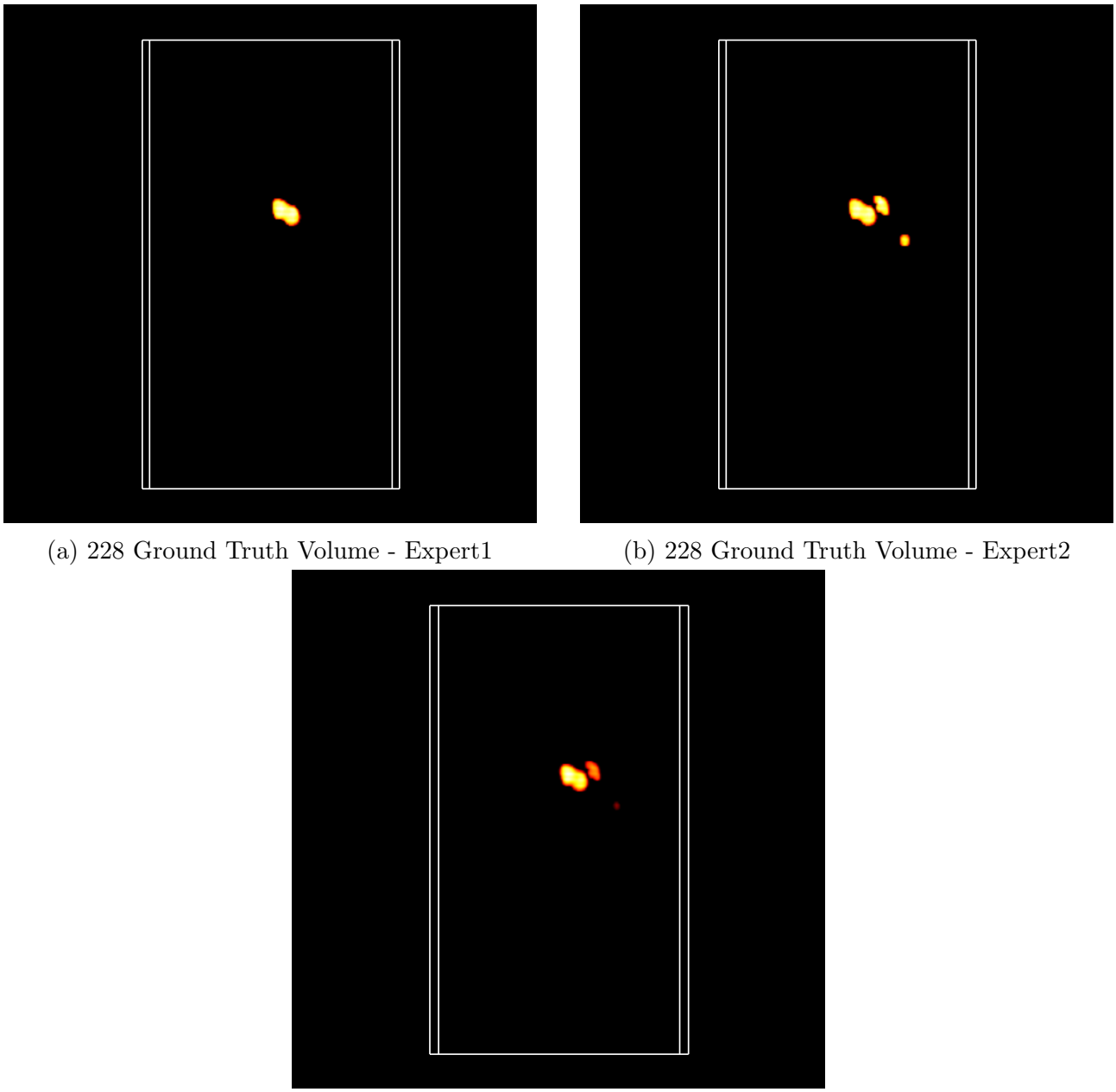


(a)



(b)

Figure 4.7: Performance Evaluation on filtered (a) Training set, and (b) Test set volumes with respect to two ground truth volumes Lung tumor images from PET scans provided by the two experts, respectively. The X-axis depicts the classifiers trained with training data samples of different thresholds of high tumor fraction, and Y-axis depicts the average F-scores obtained.

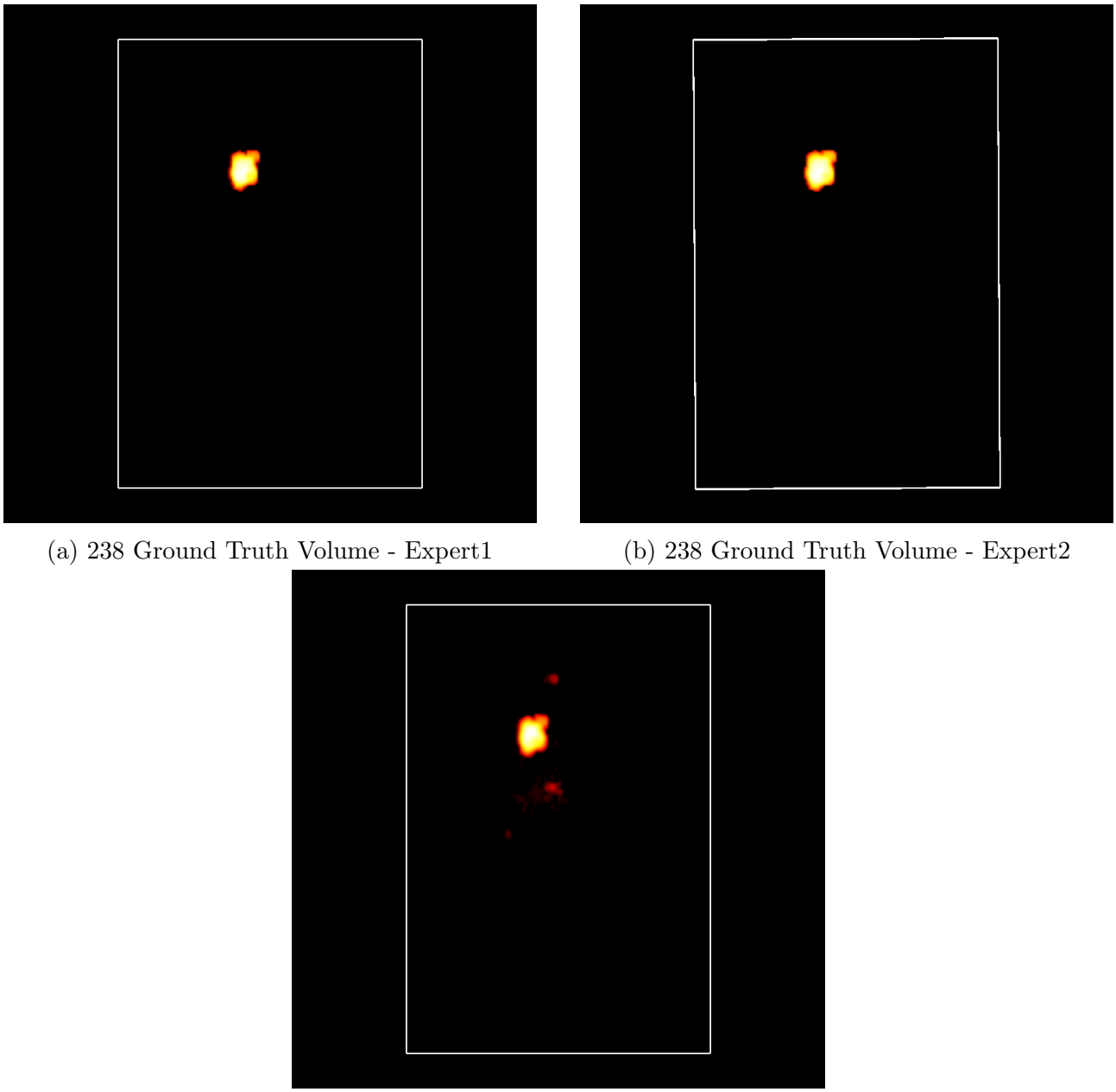


(a) 228 Ground Truth Volume - Expert1

(b) 228 Ground Truth Volume - Expert2

(c) 228 Classified Volume

Figure 4.8: The subFigure 4.8c is the visualization of classified volume 228 from test set volume with respect to two ground truth volumes using MTDemo. The subFigures 4.8a and 4.8b are ground truth volumes provided by experts 1 and 2, respectively.

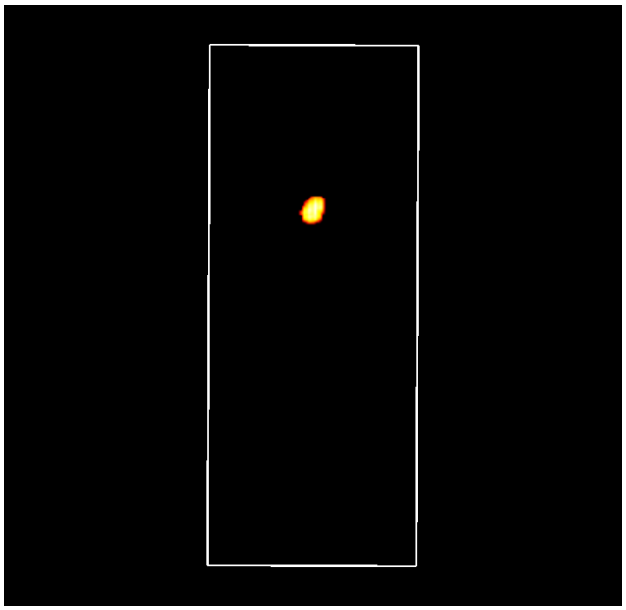


(a) 238 Ground Truth Volume - Expert1

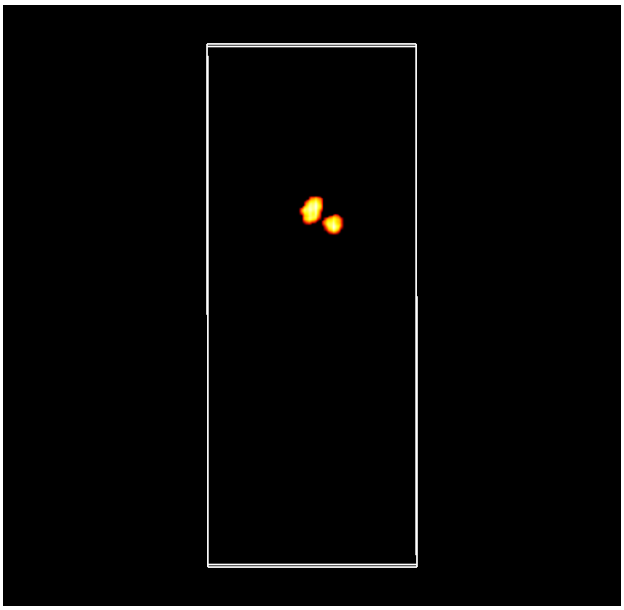
(b) 238 Ground Truth Volume - Expert2

(c) 238 Classified Volume

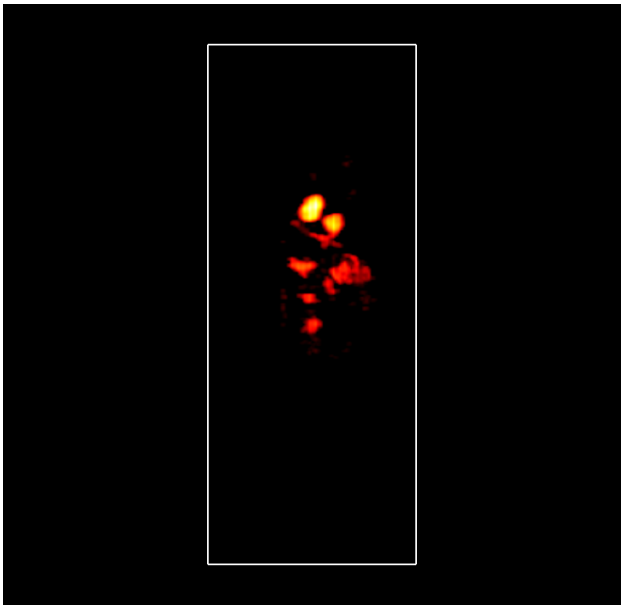
Figure 4.9: The subFigure 4.9c is the visualization of classified volume 238 from the test set volume with respect to two ground truth volumes using MTDemo. The subFigures 4.9a and 4.9b are ground truth volumes provided by experts 1 and 2, respectively.



(a) 236 Ground Truth Volume - Expert1

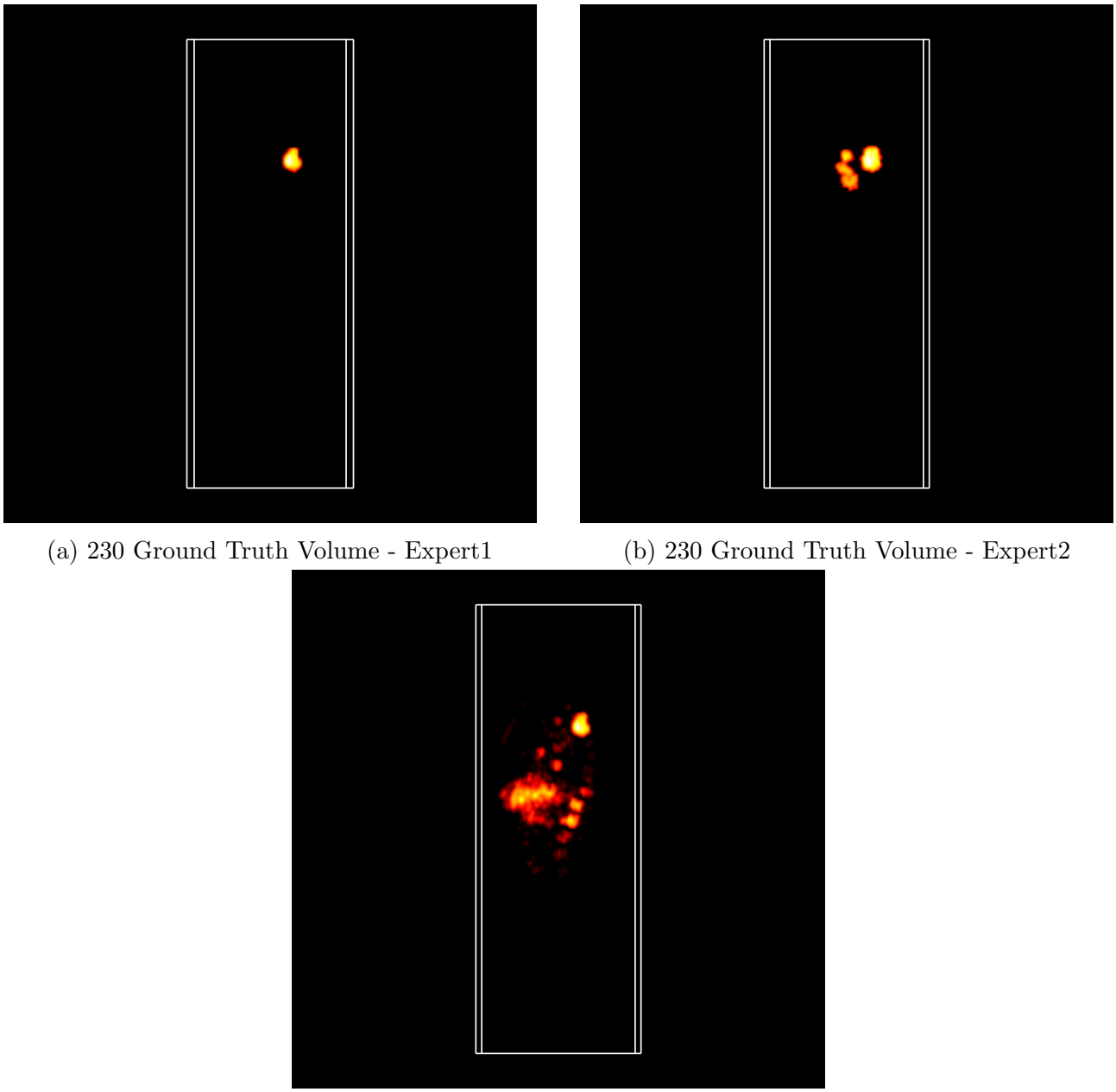


(b) 236 Ground Truth Volume - Expert2



(c) 236 Classified Volume

Figure 4.10: The subFigure 4.10c is the visualization of classified volume 236 from test set volume with respect to two ground truth volumes using MTDemo. The subFigures 4.10a and 4.10b are ground truth volumes provided by experts 1 and 2, respectively.

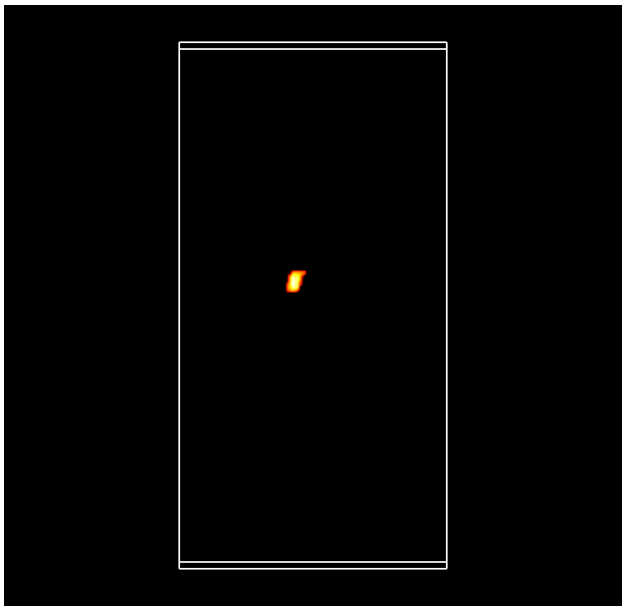


(a) 230 Ground Truth Volume - Expert1

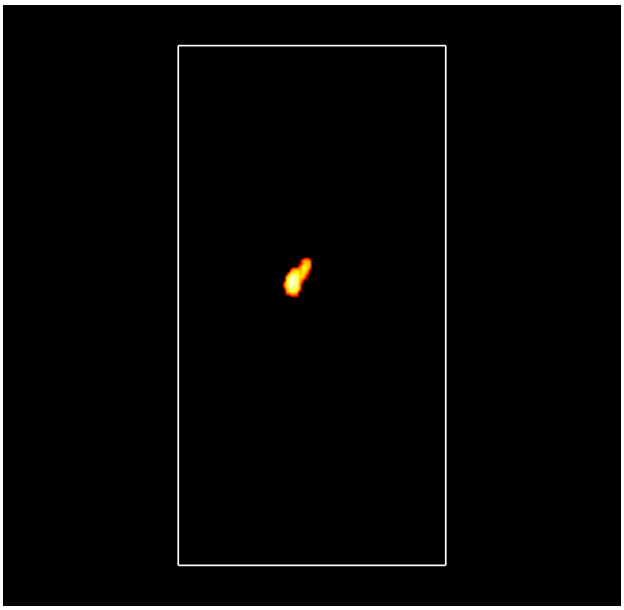
(b) 230 Ground Truth Volume - Expert2

(c) 230 Classified Volume

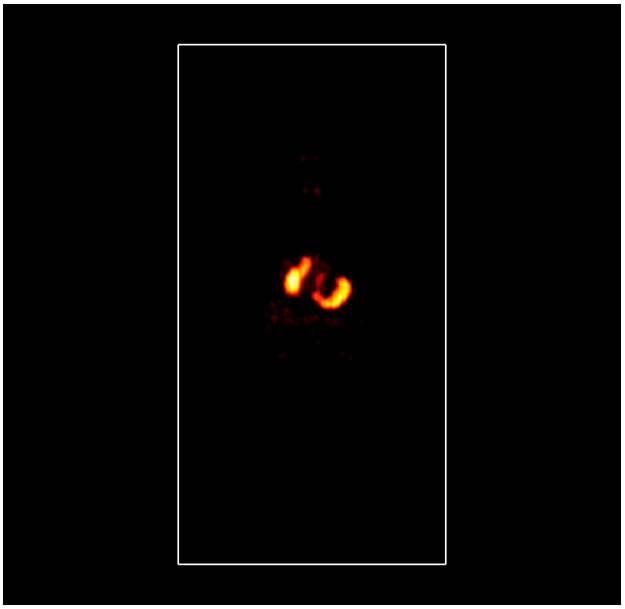
Figure 4.11: The subFigure 4.11c is the visualization of classified volume 230 from test set volume with respect to two ground truth volumes using MTDemo. The subFigures 4.11a and 4.11b are ground truth volumes provided by experts 1 and 2, respectively.



(a) 244 Ground Truth Volume - Expert1



(b) 244 Ground Truth Volume - Expert2



(c) 244 Classified Volume

Figure 4.12: The subFigure 4.12c is the visualization of classified volume 244 from the test set volume with respect to two ground truth volumes using MTDemo. The subFigures 4.12a and 4.12b are ground truth volumes provided by experts 1 and 2, respectively.

Chapter 5

Conclusion & Future work

The authors in [7] presented an exploratory tool to explore and analyze PET scan volumes' morphological features using an unsupervised learning technique, namely the Self Organising Maps (SOMs). Their work showed the possibility of automating the attribute selection from the PET scans to detect lung tumors efficiently. In this thesis, we successfully present general-purpose data science tool to analyze vector attributes in Max Tree built for each PET scan using a supervised machine learning technique, Random Forest. The method is applied to medical FDG-PET scans for lung tumor detection to examine the vector attributes in Max tree data structures and to find the possibility of detecting the feature automatically. In this work, we trained the classifier with different biased samples. When trained with higher tumor nodes, i.e., with $\geq 10\%$ of high tumor nodes, it was observed, especially with a 50% high tumor percentage of the total nodes, the false negative rate was the least, which showed when the classifier is trained with the training set with a significant number of high fraction tumor nodes .i.e, $> 10\%$ of the total nodes provided a better performance of the classifier in classifying the high tumor nodes, the same behavior was observed in work by authors [7]. Also, it was observed when the classifier was trained with a significant number of high fraction tumor nodes, i.e., $>$ 10% of the total nodes, that *Intensity Mean* turns out to be the most significant feature ranked by the classifier to detect the lung tumor effectively. We even experimented with the High tumor fraction threshold. While preprocessing, we considered minima and maximal intervals of thresholds of High tumor fraction, i.e., when tumor volume out of total volume accounted for $\geq 10\%$ to $\geq 90\%$.

On experimenting with several thresholds, the classifier trained with data samples with a higher tumor fraction threshold which are $\geq 70\%$, $\geq 80\%$, $\geq 90\%$ respectively, performed better than classifiers trained with a dataset labeled with a lower point. This is because the classifiers trained with these thresholds relied more on *Intensity Mean* features to distinguish the high tumor nodes, with this feature ranked higher by all these three trained classifiers. This provides an intriguing indication that the classifier's ability to classify high tumor nodes accurately is highly dependent on the sensitivity of the *Intensity Mean* feature. Furthermore, the classifiers trained with datasets labeled with lower thresholds showed features *Intensity Mean*, and *Centre of Mass Z* high ranked to classify the high tumor nodes. In contrast, the classifier trained with datasets labeled with High tumor fraction accounting to $\geq 90\%$ of tumor volume of the total

volume showed feature *Intensity Mean* was highly significant in classifying high tumor nodes. Also, we conclude from the results that the tumor fraction threshold affects the classifier's performance when evaluated with ground truth volumes, a classifier trained with data set with a high tumor fraction resulted in increased average f-scores. However, the overall average fscore on the test set and training set seems to be low due to the classifier's sensitivity to organs with high Intensity. This behavior was observed even in the previous work by the authors in [7], where trained SOMs turn out sensitive to certain features of PET scans, usually to high-intensity organs.

As a suggestion for future work, it may be advantageous to explore using Computed Tomography (CT) scans for lung tumor detection instead of FDG-PET scans. Although FDG-PET scans can detect lung tumors and other inflammatory cells, the presence of these cells can introduce complexity in tumor detection. In contrast, CT scans provide a clearer view of structures and their gray levels, allowing for more precise data processing and potentially improving lung tumor detection and classification accuracy. Additionally, incorporating texture-based features such as roughness may improve the visualization of false tumors and facilitate a more effective evaluation of classifier performance. To address issues of imbalanced data sets, it may be worth considering using a hybrid sampling technique that combines majority under-sampling and minority over-sampling methods. Regarding classification algorithms, boosting with a hybrid sampling technique instead of bagging may improve performance and reduce misclassification. Adaboost, which employs sequential learning with weights updated for every tree learner, is a promising option.

Bibliography

[1] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, "Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, vol. 71, no. 3, pp. 209–249, 2021.

[2] M. Malvezzi, G. Carioli, P. Bertuccio, T. Rosso, P. Boffetta, F. Levi, C. La Vecchia, and E. Negri, "European cancer mortality predictions for the year 2016 with focus on leukaemias," *Annals of Oncology*, vol. 27, no. 4, pp. 725–731, 2016.

[3] A. A. Shah, H. A. M. Malik, A. Muhammad, A. Alourani, and Z. A. Butt, "Deep learning ensemble 2d cnn approach towards the detection of lung cancer," *Scientific Reports*, vol. 13, no. 1, p. 2987, 2023.

[4] M. Desai and M. Shah, "An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (mlp) and convolutional neural network (cnn)," *Clinical eHealth*, vol. 4, pp. 1–11, 2021.

[5] M. S. Al-Tarawneh, S. Al-Habashneh, N. Shaker, W. Tarawneh, and S. Tarawneh, "Lung cancer detection using morphological segmentation and gabor filtration approaches," *International Journal of Engineering Research*, vol. 3, no. 7, 2014.

[6] R. Jones, "Connected filtering and segmentation using component trees," *Computer Vision and Image Understanding*, vol. 75, no. 3, pp. 215–228, 1999.

[7] H. Gan, S. Gazagnes, M. Babai, and M. H. F. Wilkinson, "Self-organising attribute maps and pattern spectra: Novel explorative data analysis tools for high-dimensional vector-attribute filtering," *JOURNAL OF LATEX CLASS FILES*, vol. 14, no. 8, 2022.

[8] S. Gazagnes and M. H. F. Wilkinson, "Distributed connected component filtering and analysis in 2d and 3d tera-scale data sets," *IEEE Transactions on Image Processing*, vol. 30, pp. 3664–3675, 2021.

[9] P. Salembier and M. H. F. Wilkinson, "Connected operators : A review of region-based morphological image processing techniques," *IEEE Signal Processing Magazine*, vol. 26, pp. 136 – 157, 12 2009.

[10] M. H. F. Wilkinson, "Attribute-space connected filters," in *Mathematical Morphology: 40 Years On* (C. Ronse, L. Najman, and E. Decencière, eds.), (Dordrecht), pp. 85–94, Springer Netherlands, 2005.

[11] G. K. Ouzounis, S. Giannakopoulos, C. E. Simopoulos, and M. H. F. Wilkinson, "Robust extraction of urinary stones from ct data using attribute filters," in *Proc. 16th International Conference on Image Processing 2009*, pp. 2629–2632, IEEE, 2009. Relation: http://www.rug.nl/informatica/onderzoek/bernoulli Rights: University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science.

[12] F. Kiwanuka and M. H. F. Wilkinson, "Automatic attribute threshold selection for morphological connected attribute filters," *Pattern Recognition*, vol. 53, 12 2015.

[13] F. N. Kiwanuka and M. H. Wilkinson, "Cluster based vector attribute filtering," *Mathematical Morphology-Theory and Applications*, vol. 1, no. 1, 2016.

[14] B. Naegel, N. Passat, N. Boch, and M. Kocher, "Segmentation using vector-attribute filters: Methodology and application to dermatological imaging," *ISMM 2007, 8th International Symposium on Mathematical Morphology*, vol. 1, 10 2007.

[15] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[16] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[17] T. Kohonen and P. Somervuo, "Self-organizing maps of symbol strings," *Neurocomputing*, vol. 21, no. 1-3, pp. 19–30, 1998.

[18] X. Hu, Y. Lian, X. Hu, Z. Liu, M. Wang, Y. Chen, Z. Yang, and H. Zhang, "Developing test color samples to compute color fidelity of light sources for printing matter," *Opt. Express*, vol. 29, pp. 43032–43048, Dec 2021.

[19] E. Bass, "Editorial ieee transactions on human-machine systems: Year in review for 2015," *IEEE Transactions on Human-Machine Systems*, vol. 46, pp. 1–8, 02 2016.

[20] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning," *The Stata Journal*, vol. 20, no. 1, pp. 3–29, 2020.

[21] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[22] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282 vol.1, 1995.

[23] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 10 2001.

[24] A. Arfiani and Z. Rustam, "Ovarian cancer data classification using bagging and random forest," in *AIP Conference Proceedings*, vol. 2168, p. 020046, AIP Publishing LLC, 2019.

[25] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How many trees in a random forest?," in *Machine Learning and Data Mining in Pattern Recognition: 8th International Conference, MLDM 2012, Berlin, Germany, July 13-20, 2012. Proceedings 8*, pp. 154–168, Springer, 2012.

[26] S. Janitza and R. Hornung, "On the overestimation of random forest's out-of-bag error," *PloS one*, vol. 13, p. e0201904, 2018.

[27] M. Westenberg, J. Roerdink, and M. Wilkinson, "Volumetric attribute filtering and interactive visualization using the max-tree representation," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2943–2952, 2007.

[28] H. Gan, S. Gazagnes, M. Babai, and M. H. F. Wilkinson, "Experiments and discussion," in *Journal of LaTeX Class Files*, pp. 4–6, University of Groningen, 2022. Reprinted from the journal "Journal of LaTeX Class Files" edited by Hyoyin Gan and Simon Gazagnes and Mohammad Babai and Michael H. F. Wilkinson, published in 2022.

[29] K. Kitajima, M. Nakajo, H. Kaida, R. Minamimoto, K. Hirata, M. Tsurusaki, H. Doi, Y. Ueno, K. Sofue, Y. Tamaki, *et al.*, "Present and future roles of fdg-pet/ct imaging in the management of gastrointestinal cancer: an update," *Nagoya journal of medical science*, vol. 79, no. 4, p. 527, 2017.

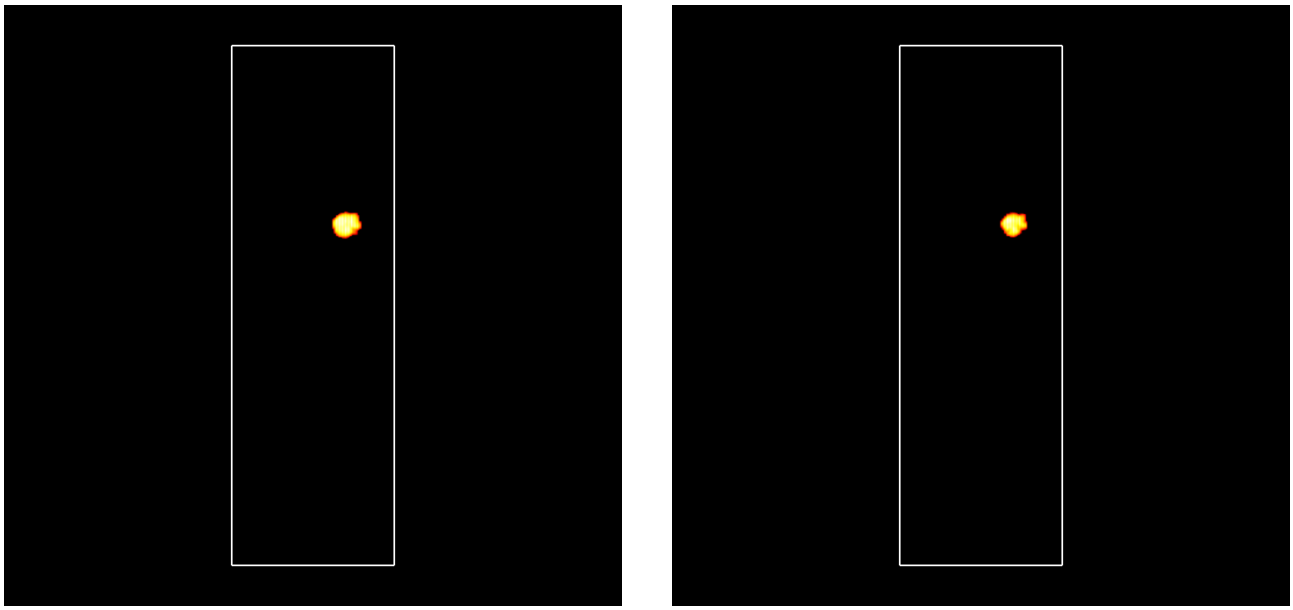
[30] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior, "The cancer imaging archive (tcia): Maintaining and operating a public information repository," *Journal of Digital Imaging*, vol. 26, pp. 1045–1057, Dec. 2013. Copyright: Copyright 2013 Elsevier B.V., All rights reserved.

[31] M. Machtay, F. Duan, and e. Siegel, Barry A., "Prediction of survival by [18f]fluorodeoxyglucose positron emission tomography in patients with locally advanced non-small-cell lung cancer undergoing definitive chemoradiation therapy: Results of the acrin 6668/rtog 0235 trial," *Journal of Clinical Oncology*, vol. 31, pp. 3823–3830, Oct. 2013.

Appendix A

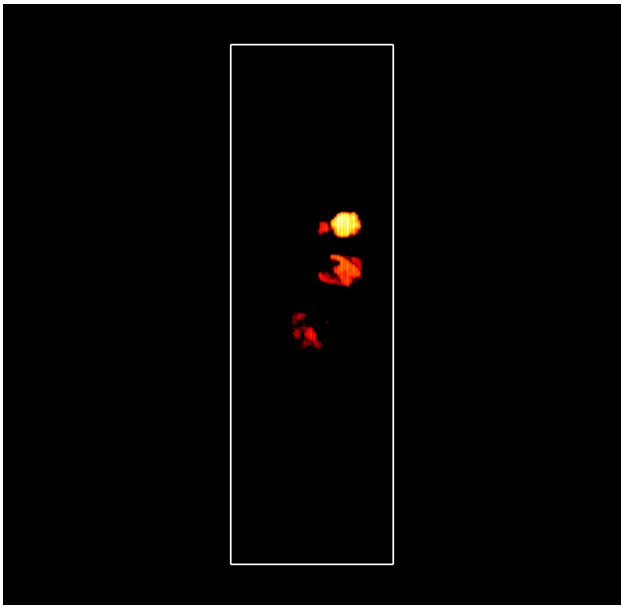
Appendices

The subFigures A.1c, A.2c, A.3c, and A.4c demonstrate the visualization of classified volumes 046, 074, 104, and 105 from the training set volumes. The performance of these volumes was evaluated using the f-score metric, resulting in values of 0.4, 0.3, 0.2, and 0.45 when compared to ground truth volumes provided by expert 1 and values of 0.6, 0.3, 0.1, and 0.2 when compared to ground truth volumes provided by expert 2. Additionally, the subFigures A.5c, A.7c, A.6c, and A.8c demonstrate the visualization of classified volumes 225, 232, 249, and 241 from the test set volumes. The f-score metric was used to evaluate the performance of these volumes, resulting in values of 0.3, 0.4, 0.24, and 0.2 when compared to ground truth volumes provided by expert 1 and values of 0.1, 0.4, 0.3, and 0.2 when compared to ground truth volumes provided by expert 2. In these volumes, the classifier identifies organs like the brain, stomach, and large intestine as tumors, and this is due to the classifier's sensitivity to *Intensity Mean* feature of the nodes, which are classified as false tumor nodes by the classifier.



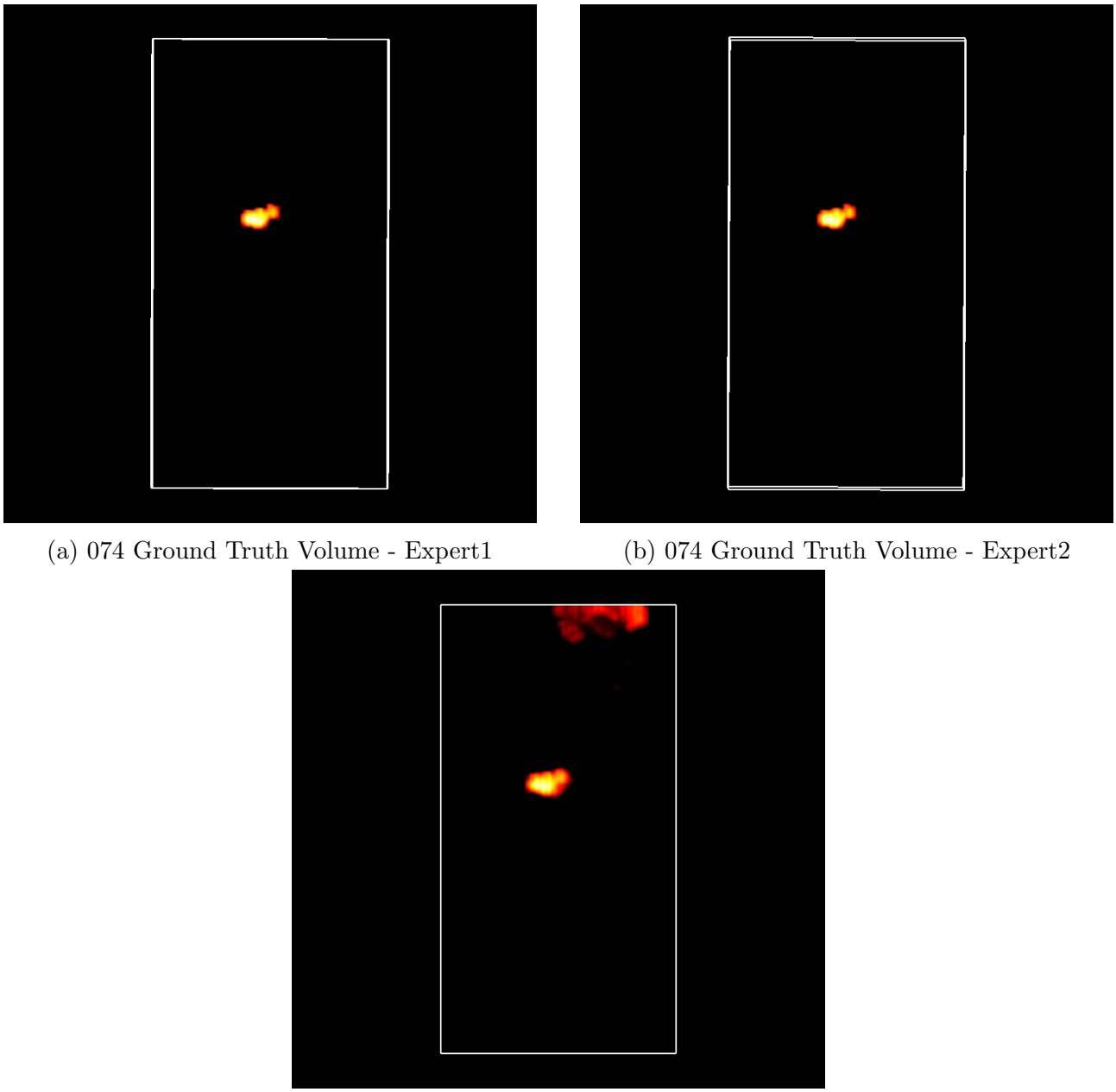
(a) 046 Ground Truth Volume - Expert1

(b) 046 Ground Truth Volume - Expert2



(c) 046 Classified Volume

Figure A.1: The subFigure A.1c is the visualization of classified volume 046 from the training set volume with respect to two ground truth volumes using MTdemo. The subFigures A.1a and A.1b are ground truth volumes provided by experts 1 and 2, respectively.

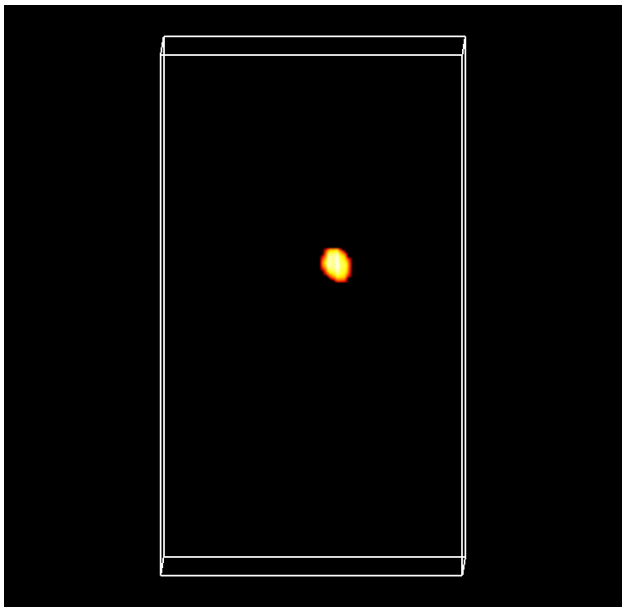


(a) 074 Ground Truth Volume - Expert1

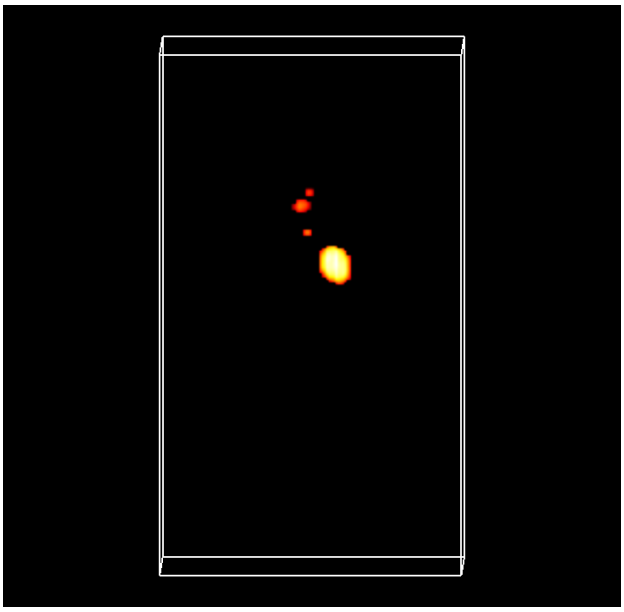
(b) 074 Ground Truth Volume - Expert2

(c) 074 Classified Volume

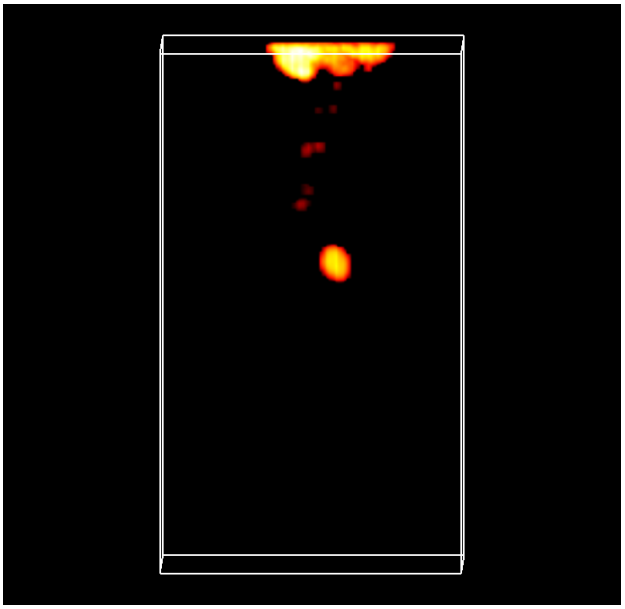
Figure A.2: The subFigure A.2c is the visualization of classified volume 074 from the training set volume with respect to two ground truth volumes using MTdemo. The subFigures A.2a and A.2b are ground truth volumes provided by experts 1 and 2, respectively.



(a) 104 Ground Truth Volume - Expert1

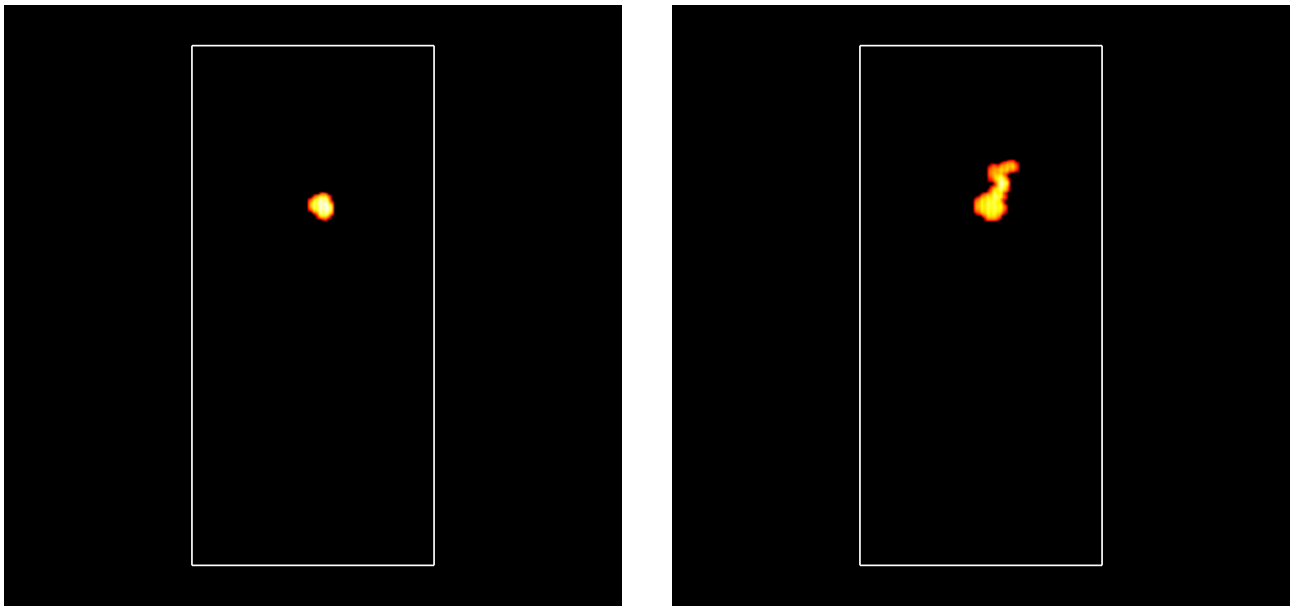


(b) 104 Ground Truth Volume - Expert2



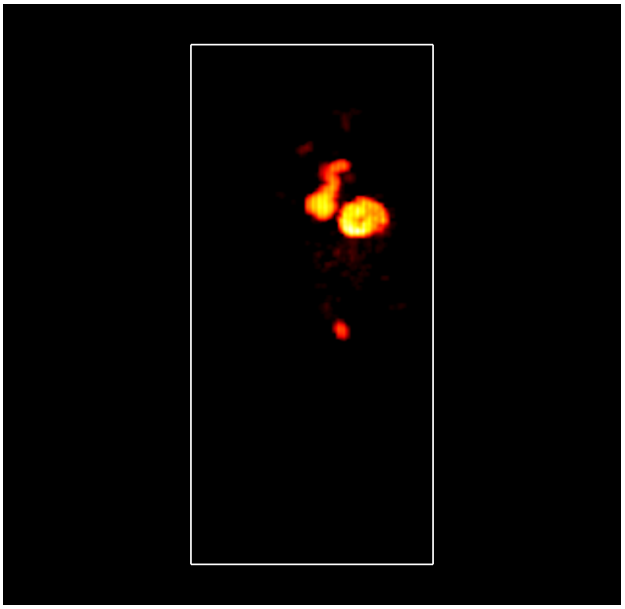
(c) 104 Classified Volume

Figure A.3: The subFigure A.3c is the visualization of classified volume 104 from the training set volume with respect to two ground truth volumes using MTdemo. The subFigures A.3a and A.3b are ground truth volumes provided by experts 1 and 2, respectively.



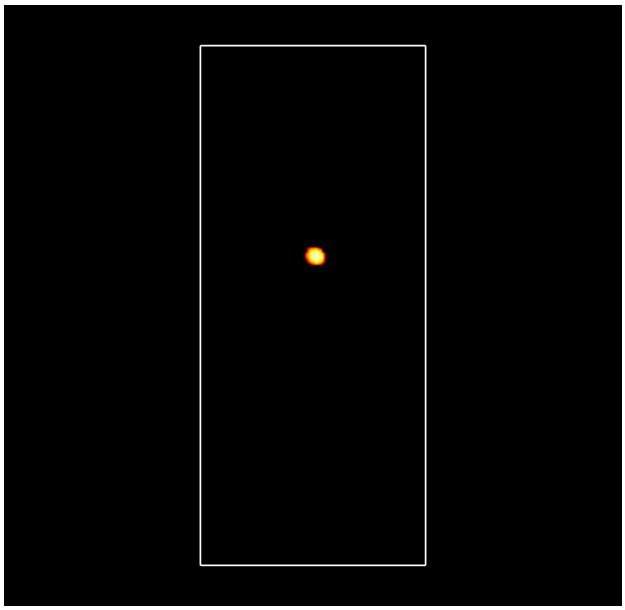
(a) 105 Ground Truth Volume - Expert1

(b) 105 Ground Truth Volume - Expert2

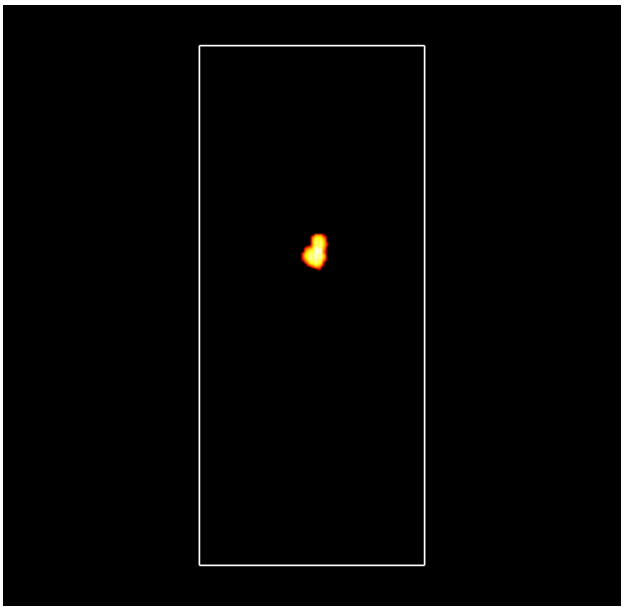


(c) 105 Classified Volume

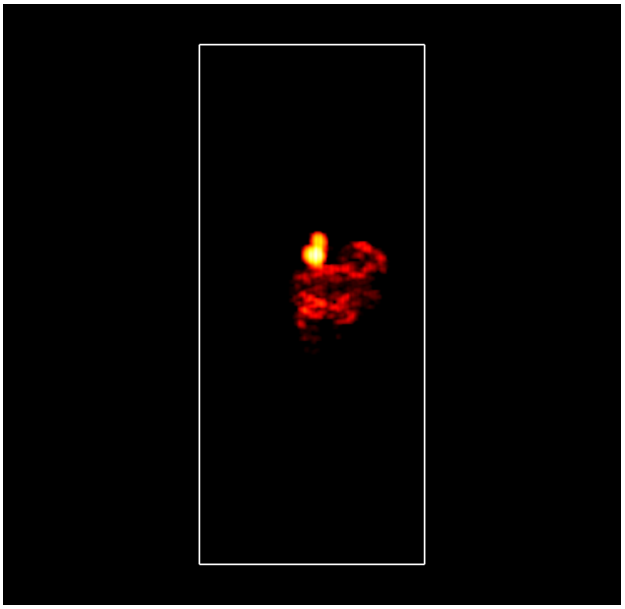
Figure A.4: The subFigure A.4c is the visualization of classified volume 105 from the training set volume with respect to two ground truth volumes using MTdemo. The subFigures A.4a and A.4b are ground truth volumes provided by experts 1 and 2, respectively.



(a) 225 Ground Truth Volume - Expert1

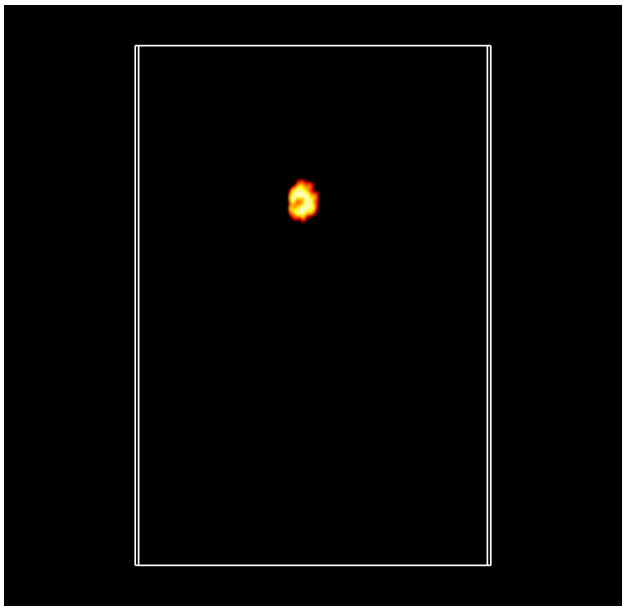


(b) 225 Ground Truth Volume - Expert2

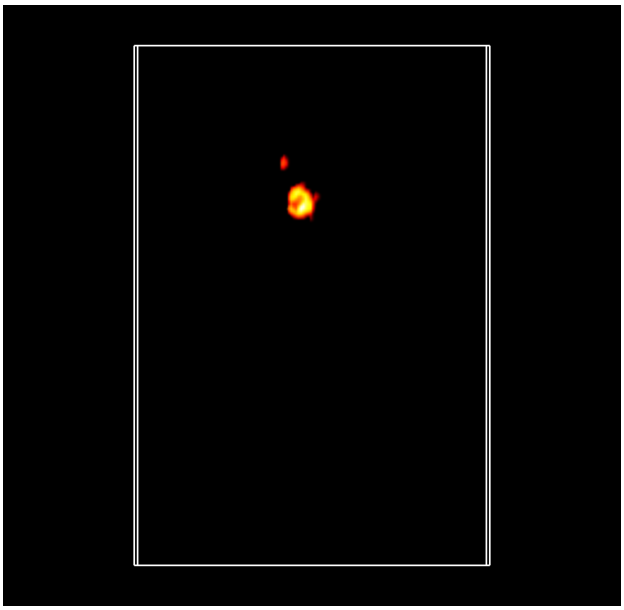


(c) 225 Classified Volume

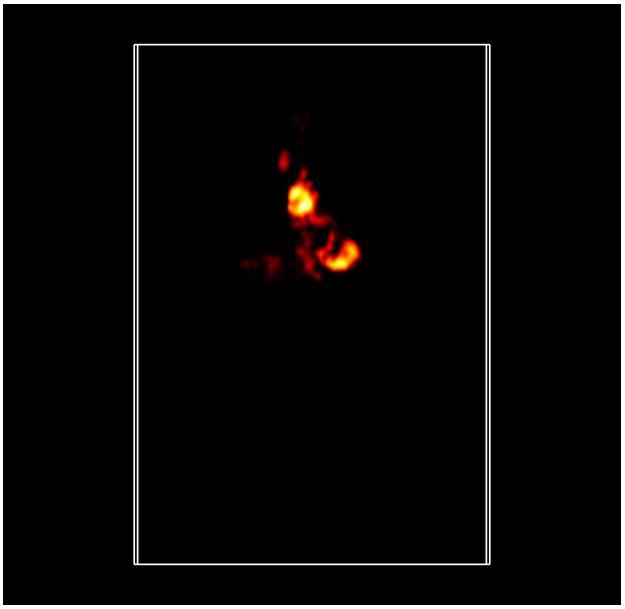
Figure A.5: The subFigure A.5c is the visualization of classified volume 225 from the test set volumes with respect to two ground truth volumes using MTdemo. The subFigures A.5a and A.5b are ground truth volumes provided by experts 1 and 2, respectively.



(a) 249 Ground Truth Volume - Expert1

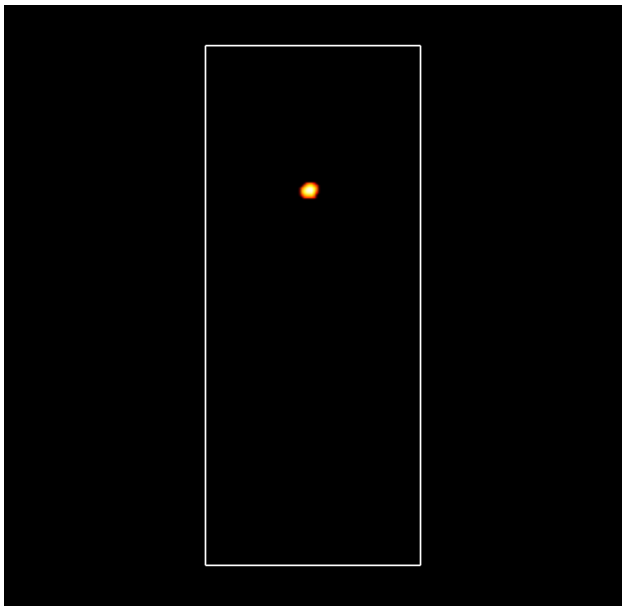


(b) 249 Ground Truth Volume - Expert2

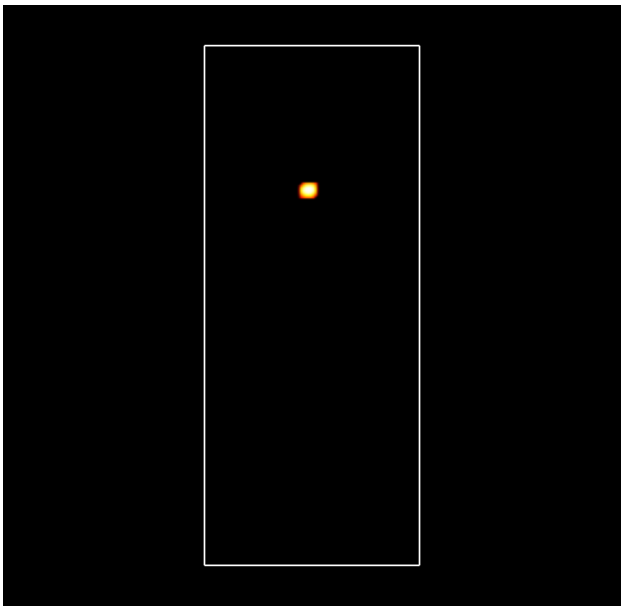


(c) 249 Classified Volume

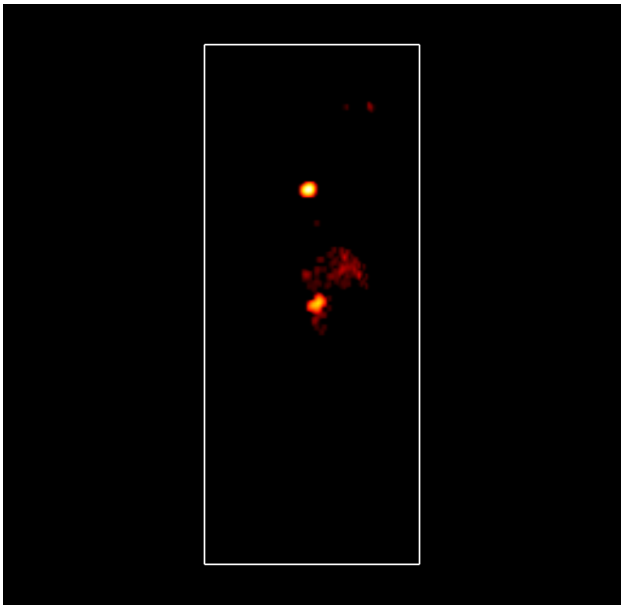
Figure A.6: The subFigure A.6c is the visualization of classified volume 249 from the test set volumes with respect to two ground truth volumes using MTdemo. The subFigures A.6a and A.6b are ground truth volumes provided by experts 1 and 2, respectively.



(a) 232 Ground Truth Volume - Expert1

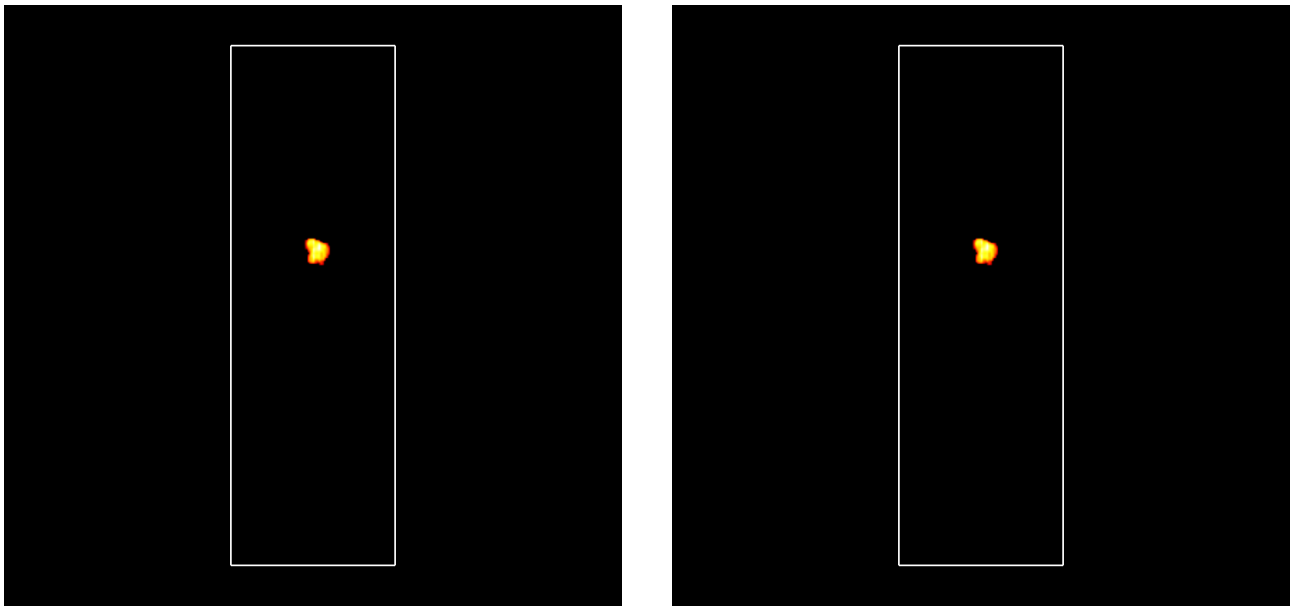


(b) 232 Ground Truth Volume - Expert2



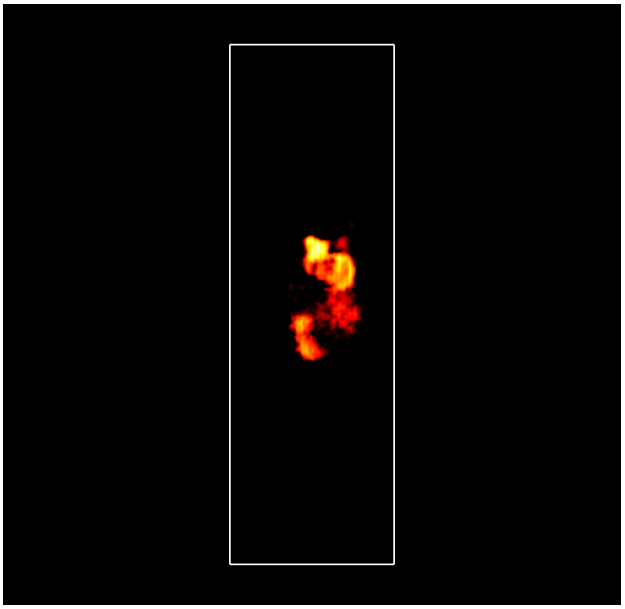
(c) 232 Classified Volume

Figure A.7: The subFigure A.7c is the visualization of classified volume 232 from the test set volumes with respect to two ground truth volumes using MTdemo. The subFigures A.7a and A.7b are ground truth volumes provided by experts 1 and 2, respectively.



(a) 241 Ground Truth Volume - Expert1

(b) 241 Ground Truth Volume - Expert2



(c) 241 Classified Volume

Figure A.8: The subFigure A.8c is the visualization of classified volume 241 from the test set volumes with respect to two ground truth volumes using MTdemo. The subFigures A.8a and A.8b are ground truth volumes provided by experts 1 and 2, respectively.