LUNG CANCER SEGMENTATION OF 3D CONE BEAM COMPUTED TOMOGRAPHY SCANS

Somnath Basu Roy Chowdhury

LUNG CANCER SEGMENTATION OF 3D CONE BEAM COMPUTED TOMOGRAPHY SCANS

Project Report submitted to
Indian Institute of Technology Kharagpur
for the Award of the Degree

of

Dual Degree (B.Tech (Hons.) + M.Tech)
in Electrical Engineering
with specialization in Instrumentation and Signal Processing

by

Somnath Basu Roy Chowdhury
Roll No: 13EE35017

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
APRIL 2018

DECLARATION

I certify that

(a) the work contained in this project report is original and has been done by me under the guidance of my supervisor(s).

(b) the work has not been submitted to any other Institute for any degree or diploma.

(c) I have followed the guidelines provided by the Institute in preparing the project report.

(d) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(e) whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the project report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Signature of the Student

CERTIFICATE

This is to certify that the project report entitled **Lung Cancer Segmentation of 3D Cone Beam Computed Tomography**, submitted by **Somnath Basu Roy Chowdhury** to Indian Institute of Technology, Kharagpur, is a record of bona fide research work under my (our) supervision and is worthy of consideration for the award of the degree of Dual Degree in Instrumentation and Signal Processing of the Institute.

Prof. Jayanta Mukhopadhyay

Prof. Siddhartha Sen

Date:

ACKNOWLEDGEMENT

I am highly indebted to Professor Jayanta Mukhopadhyay, Department of Computer Science and Engineering for his guidance and constant supervision as well as for providing necessary information regarding the project and also for his support in helping me successfully completing the project. I would also like to thank Mr. Bijju Kranthi Veduruparthi, for his constant guidance during the project. I am also grateful to Professor Siddhartha Sen, for his help and guidance relating to the project. This work has been supported by Ministry of Human Resource Development (MHRD), India. I also express my gratitude to Tata Medical Center, Kolkata for providing the CBCT data of lung scans.

Contents

List of Figures

List of Tables

Abstract

Lung cancer is one of the most prevalent form of cancer, causing an alarming number of deaths worldwide and significant amount of healthcare costs. It is also a very harmful form of cancer; overall only 17% of the people diagnosed with lung cancer in US survive five years after the diagnosis and the rate is even lower in lesser developed countries. Stages of lung cancer depend on the area it has metastasized. Current diagnosis techniques involve imaging and low-dose Computed Tomography (CT) scans for cancer detection and segmentation. Apart from early diagnosis, automated segmentation by Computer-Aided Diagnosis (CAD) systems are helpful in studying whether a medication for the cancer is producing desirable results. However, this has several disadvantages as frequent high dosage CT scans are detrimental to a person's health. Cone Beam Computed Tomography (CBCT) scans are low dosage CT scans where scans suffer from several artifacts and noise. The motivation of our work is to formulate a CAD system which is able to segment cancer from 3D CBCT scans.

Segmentation techniques are extensively studied in image processing and pattern recognition based literature. Image segmentation is widely studied in medical domain for diagnosis of various diseases. CBCT is an inherently noisy image domain which makes the segmentation process challenging and conventional machine learning techniques fail to achieve acceptable performance in this domain. We hypothesized two major reasons for this behavior. First, intensity of the cancer cells is almost similar to the soft tissues which makes it difficult for the convolutional layers to discern. Second, the high class imbalance in the dataset makes the network biased towards the class having more instances. In this work, we present a novel technique of lung cancer localization and segmentation in 3D Cone Beam Computed Tomography (CBCT) scans. In our work, we have independently tackled the problem of localization and cancer segmentation, and achieved significant improvement in pixel-wise segmentation over conventional learning techniques.

Chapter 1

Introduction

Lung cancer is one of the most prevalent form of cancer, which causes a large number of deaths and leads to significant health-care costs worldwide. Computed Tomography (CT) scans are essential for regular diagnosis of the disease. Lung cancer treatment involves several stages of medication and analysis to determine whether a particular medication is working or not, thereby requiring a patient to undergo several CT scans. Researchers have found that frequent CT scans may have hazardous effects on an individual's health and even increase the risk of developing cancer. Physicians are therefore advised to recommend CT scans only when the benefits from the scan in diagnosis overweigh the risks associated with it. Low-dosage Cone-beam Computed Tomography (CBCT) [1] scan is an alternative used for patients who require frequent scans for planning of the cancer treatment. It has been proposed that if small lung cancer cells are detected and removed at an early stage, the survival-rate can be increased by a significantly.
Segmentation in low resolution and noisy CBCT scans is a tedious job and requires extensive manual supervision. Even highly trained radiation oncologists often make mistakes marking the benign regions as cancerous due to the similarity of the pixel intensity values. Manual segmentation overestimates the lesion region in order to make sure the whole cancerous volume has been segmented [2]. Thereby, the results also suffer from high false positive and false negative errors.

Misdiagnosis is accompanied by either delay in treatment or unnecessary treatment costs piling up health-care expenditures. Keeping this challenges in mind, we propose to develop a cancer detection system to assist radiologists in cancer segmentation. In recent years, medical imaging using deep learning architectures have gained popularity due to its capability to produce state-of-the-art results and scalability to large number of training data. We implement various deep-learning based segmentation architectures and study the performance on our dataset.

The problem of cancer segmentation in CBCT is intrinsically challenging and as we observed in our experiments even state-of-the-art deep learning models like U-Net [3], fail to discern the cancer location. This arises mainly due to the similar intensity range between tumor and soft tissue region, and also due to the presence of various types of artifacts during CBCT scan capturing. Deep networks as observed in our experiments are unable to process both location information and pixel intensity simultaneously to localize the tumor region. Another disadvantage of deep networks is that they require large amount of training data samples. Labeled CBCT scan data is extremely difficult to gather and takes a lot of manual labor. To tackle this problem we introduce a lung cancer localization module which leverages a cascade of image processing techniques to localize the possible cancer region. The localization step is followed by feature selection, pixel-level classification and post-processing for false positive suppression. We have shown experiments with a variety of feature descriptors and obtained significant improvement in pixel-level segmentation accuracy over contemporary methods.

Keywords: Cone Beam Computed Tomography, Deep Learning, Template Mactching, Pixel-wise segmentation, False Positive Suppression.

Background

CBCT [1] is a relatively modern technology. CBCT [1] was initially developed for angiography, but with upcoming medical developments it has also included mammography and radiotherapy guidance. The cone-beam geometry was developed as an alternative to conventional CT which uses fan-beam or spiral-scan geometries, in order to provide fast acquisition of the entire Field of View (FOV). It uses a relatively less expensive radiation technology. Advantages of CBCT over conventional CT, involves a shorter scan time, robustness to image sharpness reduction caused by translation of patient, reduced distortion due to internal patient movements, and enhanced efficiency of x-ray tube. However, CBCT is a low dosage image acquisition technique leading to lower x-ray penetration. It is thereby accompanied by artifacts (like beam hardening, scatter artifacts etc) which distorts the image quality making it much more prone to acquisition noise than conventional CT. An image artifact is defined as a visualized structure in the reconstructed acquisition data that is not actually present in the subject under study. These are induced by discrepancies between the actual physical conditions of the measuring setup and the simplified theoretical assumptions used for 3D reconstruction. The most predominant artifacts are noise (arising from round off errors, electrical noise etc.), scatter (caused by photons diffracted from the designated trajectory), extinction artifacts (from missing values due to highly absorbing material), beam hardening artifacts (x-ray absorption), aliasing artifacts (occurs due to under-sampling), ring artifacts (miscalibration of detector) and motion artifacts (due to object movement under scan).

Since the late 1990s PCs have been equipped for computational unpredictability and x-ray beam tubes fit for constant exposure. This has empowered low-cost clinical frameworks to be manufactured and sufficiently compact to be used in the clinic. There exists two additional factors which have accelerated the functioning of CBCT [1] in clinics to be possible.

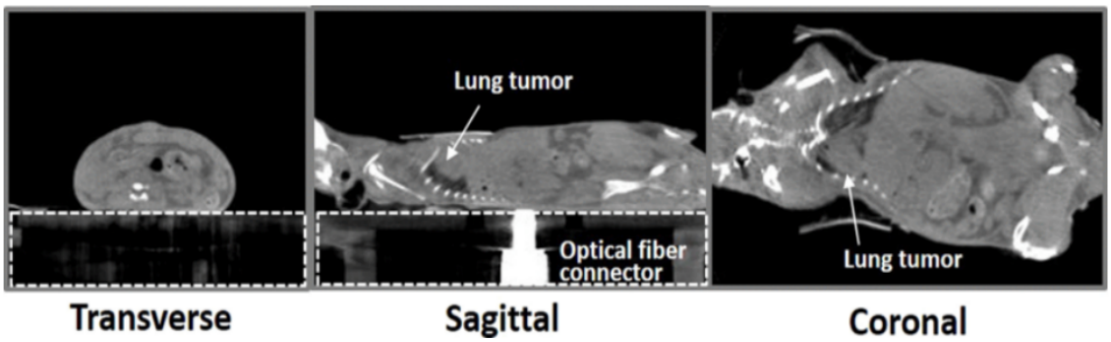


Figure 1.1: Scan positions and the corresponding location of lung cancer along (a) Tranverse (b) Sagittal and (c) Coronal plane.

Current cone-beam systems scan subjects in three conceivable positions: (1) sitting, (2) standing, and (3) prostrate as shown in Fig. 1.1. For lung cancer detection, the transverse position is chosen as it has maximum visibility of the cancer area. Hardware that requires the patient to lie recumbent physically possesses a bigger surface zone or physical impression and may not be accessible for patients with physical inabilities. The measurements of the FOV or output volume ready to be secured depend essentially on the indicator size and shape, the pillar projection geometry, and the capacity to collimate the shaft. The state of the output volume can be either barrel shaped or spherical.

The four major components of CBCT image production are (1) acquisition configuration, (2) image detection, (3) image reconstruction, and (4) image display. The image generation and detection specifications of currently available systems reflect proprietary variations in these parameters.

Artifacts and Noise

We have already defined artifacts in Section 1.1 and delineated the fundamental cause behind formation of artifacts during CBCT acquisition process. In addition to these, noise and scatter are also known to produce additional artifacts. In the following section, we will briefly summarize the information available in literature for various types of artifacts. It should be emphasized that artifacts often represent itself by the presence of streaks, line structures and shadows orientated along

projection lines. Although in the following section we try to discern between the causes of streak-like artifacts, it should be noted that many of the artifact-causing factors mentioned here have a streak-like appearance. The most predominant artifacts are [4]:

1. *Noise*: There are two main categories of noise which are considered in the reconstructed images: One type has 22 additive noise arising from round-off errors or electrical noise, and photon-count noise (quantum noise) that should be expected to follow a Poisson distribution.

2. *Scatter*: Scatter is caused by those photons that are diffracted from their unique way after association with matter. This extra share of scattered X-beams brings about expanded estimated intensities, since the scattered powers just add to the essential intensity.

3. *Extinction Artifacts*: These are often termed missing value artefacts. If the object under study contains highly absorbing material, e.g. prosthetic gold restorations, then the signal I_p recorded in the detector pixels behind that material might be near zero or really zero.

4. *Beam Hardening Artifacts*: The low energetic beams of the polychromatic range radiated by the X-beam source may endure significant assimilation when going through the patient body under examination. The more thick the last mentioned and the higher the nuclear number it is made out of, the bigger the offer of consumed wavelengths.

5. *Aliasing Artifacts*: In CBCT imaging, the sampling frequency is represented by the number of pixels per area, i.e. the pixel size of the detector. The dimensions of the detector components causes associating artefacts attributable to under-sampling.

6. *Ring Artifacts*: These are induced by deformities or uncalibrated detector components. Inferable from the roundabout direction and the sampling pro-

cess, these irregularities show up as rings in the planes coplanar with the movement plane of the source (axial planes in CBCT).

7. *Motion Artifacts*: When a patient moves amid the scanning procedure, the reconstruction does not represent that move since no data on the development is coordinated in the reproduction procedure. Henceforth, the lines along which the backprojection happens don't relate to the lines along which the lessening had been recorded, essentially in light of the fact that the question has moved amid the scanning process.

Related Work

Most of the work in literature involving lung cancer segmentation is focused on CT scans, where the image quality is better than CBCT as it involves less artifacts. In this section, we focus on the techniques utilized for tumor segmentation in CT scans and their performance. [5] developed a technique for lung tumor classification through 2D or 3D feature selection. Their work showed performance of various morphological, texture, geometric and intensity based features using conventional classification schemes. Single click ensemble segmentation [6] introduced a semi-autonomous tumor segmentation technique where the tumor region grew from an initial seed point. Initial seeds were generated by the algorithm and also provided manually. [7] presented a semi-autonomous segmentation technique where the region of interest needs to be manually fed, and segmentation is done using image processing techniques like active contours, watershed algorithm and markov chains. [8] uses prior functional knowledge from corresponding Positron Emission Tomography (PET) images and intensity knowledge from low contrast CT scans. The prior knowledge was instrumental in automatic seed selection and to enhance the random walk algorithm to aid the segmentation process. [9] introduced a novel end-to-end four step algorithm for automatic lung tumor segmentation. At first, a sequence of registration methods were deployed to transform

the tumor delineation in the follow-up scans, a statistical based model is then used to produce a segmentation map using this prior and the third step detects any leaks of tumor segment with soft tissues. The final step involves tumor boundary refinement to tackle with various noise.

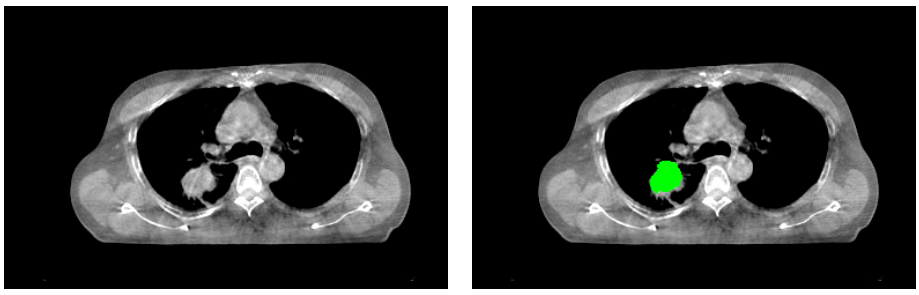
There are several other works apart from cancer segmentation using classical learning methods in CT scan domain. [10] tackled the problem of classifying whether solitary pulmonary nodule (SPN) often present in CT scans, are malignant or benign. They used co-occurence matrix based texture features for prediction. [11] is another work on SPN classification, which introduces 14 gray level co-occurence matrices followed by multilevel binomial logistic model for prediction.

There are several techniques for lung cancer classification involving machine learning techniques but all of them operate in CT domain, we will discuss few of the methods here. [12] was one of the earlier approaches towards classifying lung nodules as malignant or benign. They collected morphological features and patient's health-related data in order to classify the nodules using Artificial Neural Network (ANN) and logistic regression. [13] proposed an efficient lung nodule detection mechanism to prevent data loss by using 3D feature vectors with SVMs. [14] used CT scans to predict the presence of lung cancer, it involves segmentation of lung region by morphological techniques followed by classification using neural networks. [15] studied performance of various classification algorithms towards lung cancer classification in a similar setup as [14] but for multi-dimensional data samples. In our work we aim to perform pixel-wise segmentation of lung cancer region in a more noisy domain of CBCT, we take inspiration from some of the above works to separate the lung region using morphological approaches. The output is then forwarded to the segmentation which involves pixel-wise segmentation using learning techniques.

Chapter 2

Deep Learning Approaches

The overall aim of this work is to obtain pixel-wise segmentation of the 3D CBCT images. The 3D data consists of 54 or 64 slices of 2D data with samples taken from the lateral body position. It has been observed that out of the 54/64 slices of 2D data only around 15 images have labeled tumor region. As already discussed the segmentation procedure is a challenging task in CBCT due to the noisy domain. An example of the 2D CBCT data is shown in Fig. 2.1, where the tumor region in the original scan is barely visible. Deep learning techniques have gained a lot of popularity due to its enhanced performance in numerous challenging tasks. In this chapter we will discuss the various deep learning approaches and study the efficacy of these techniques.



(a) (b)

Figure 2.1: (a) Two-dimensional CBCT slice of a 3D scan and (b) segmentation ground-truth of the corresponding 2D slice

Initial Approach

In this section, we will discuss the different techniques/architectures implemented and the consequent problems faced in application of the deep neural network in pixel wise cancer segmentation of 3D CBCT scan. In this task we have used the neural network architecture of U-Net [3] which has served as a benchmark performance for many biomedical datasets. Our second approach was to form coarser classification of the lung region and further segmentation in the proposed regions. We used region proposal networks for the purpose of extracting likely cancer from the lung CBCT scan. In this section, we will discuss the performance of such networks like Single Shot Detector (SSD) [16] on our dataset.

U-Net Architecture

U-Net [3] is a neural network architecture which is used for pixel-wise classification of medical and astronomical data. U-Net [3] currently has been found to outperform several other architectures in the domain of medical image processing. This network architecture is useful in segmentation of data from other domain like astronomical data. U-Net uses feature maps involving various convolutional filters in order to form the final prediction, therefore is efficient in segmenting even finer pixel intensity changes.

Network Architecture

The schematic diagram of the network architecture is shown in Fig. 2.2. On the left side it consists of a contracting path and an expansive path on the right side. The contracting path follows the generic architecture of a convolutional network. It consists of the repeated application of two 3×3 convolutions kernel filters, each followed by a rectified linear unit (ReLU) [17] and a 2×2 max pooling with stride length of 2 for downsampling. At each downsampling step we double the number

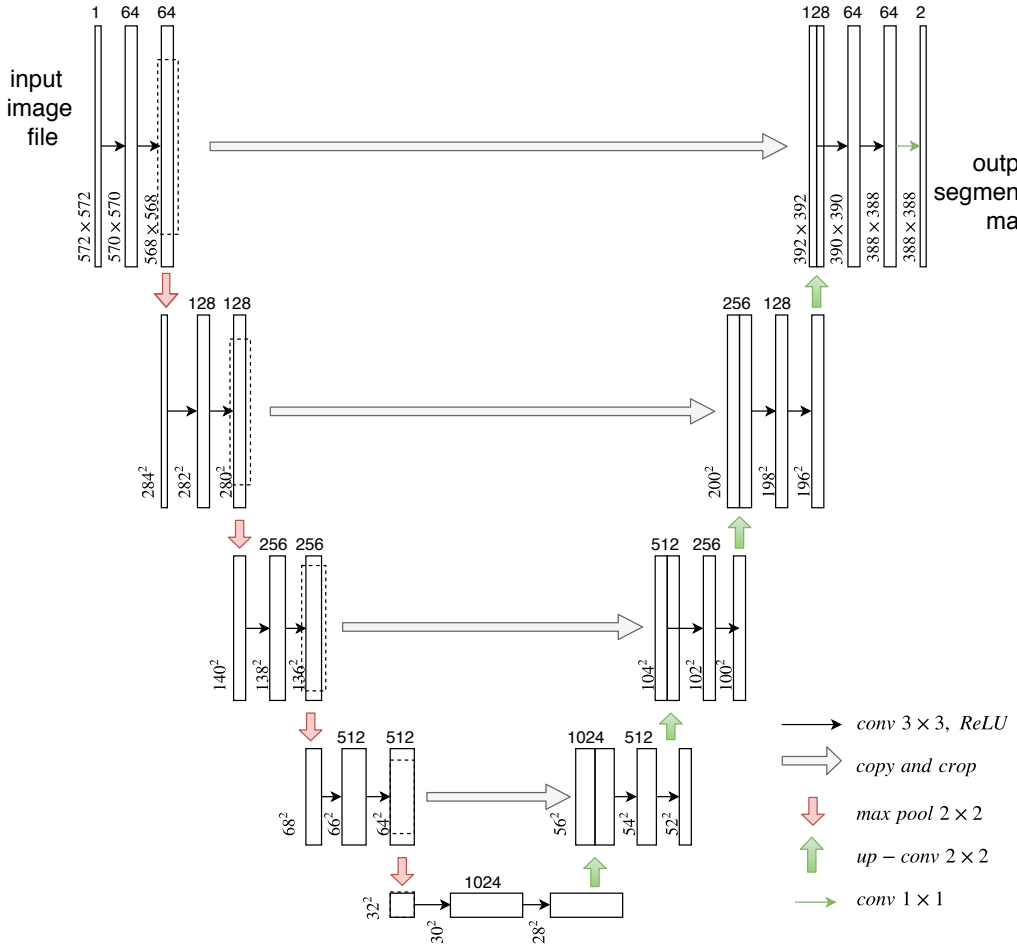


Figure 2.2: Schematic diagram of U-Net architecture

of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2×2 convolution (up-convolution) that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3×3 convolutions, each followed by a ReLU [17]. The cropping is required to compensate for the loss of border pixels in every convolution step. At the final layer a 1×1 convolution is used to map each 64-component feature vector to the desired number of classes. In total the network has 23 convolutional layers. To allow a seamless tiling of the output segmentation map, it is important to select the input tile size such that all 2×2 max-pooling operations are applied to a layer with an even x-size and y-size.

The conventional convolutional network architecture has been extended to make it

work with very few training images and yields more precise segmentations as shown in Fig. 2.2. The main idea in is to supplement a usual contracting network by successive layers, where pooling operators are replaced by upsampling operators. Hence, these layers increase the resolution of the output. In order to localize, high resolution features from the contracting path are combined with the upsampled output. A successive convolution layer can then learn to reconstruct a more precise output using this information.

Loss Function

We have mentioned earlier that in a single 3D CBCT scan only about 15 slices has cancer region present in it. This leads to a high class imbalance in the dataset. Morever in a single scan the ratio of non-zero and null valued pixels is on average 100:1. Due to this imbalance the network was returning completely null segmentation maps, in order to tackle this issue, we use loss functions which penalizes the network heavily for classifying the cancer pixels wrong. The two loss function used for the experiments were (1) Dice Coefficient Loss [18] (2) Weighted Cross-Entropy. In this section the noise and artifacts were also taken into account using median filtering as a preprocessing step.

1. **Dice Coefficient**. The SorensenDice Index or Dice Coefficient [18], is a statistical metric used for comparing the similarity of two samples. It is given as

$$\mathbf{DSC} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

   where, TP, FP and FN denote the true positive, false positive and false negative rates respectively.

2. **Weighted Cross-Entropy**. The cross entropy between two probability distributions p and q over the same underlying set of samples X calculates the average number of bits needed to identify a sample drawn from the set, if a coding scheme is used that is optimized for an "unnatural" probability

distribution q, rather than the "true" distribution p. For binary pixel-wise classification the weights are $[w_0, w_1]$. In our experiments the weights used were $[1, 1000]$. For a label y and predicted label $\hat{y}$, the cross-entropy loss is defined as

$$\mathbf{CE} = \sum w_1 y \log(\hat{y}) + w_0(1-y)\log(1-\hat{y})$$

The above equation for cross entropy is applicable for binary labels, but this loss function can be extended to include multiple classes in case of multiclass classification problems.

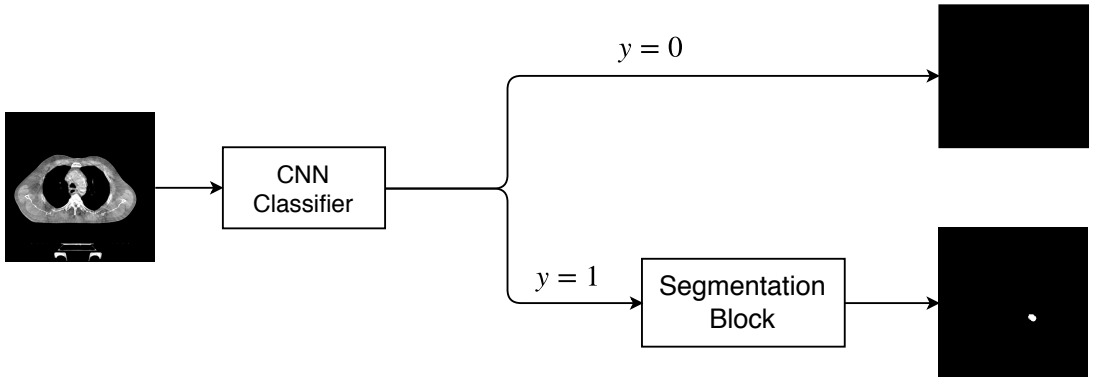


Figure 2.3: Schematic diagram of the 2-step segmentation scheme. CNN classifier used in this scheme is either VGGNet/ AlexNet. U-Net is used as the segmentation block.

Two-step Segmentation Scheme

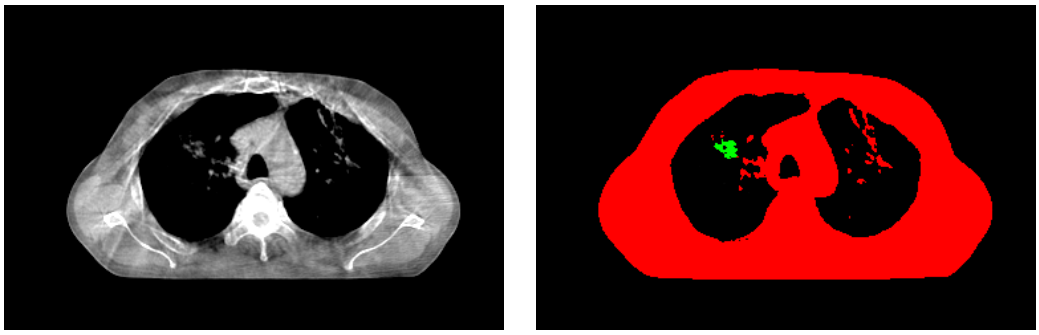
In the last section, we discussed the issue regarding class imbalance between black and white pixels in our dataset. Our solution of incorporting loss functions is not fruitful if most of the segmentation output are null (only about 15 out of 54 slices have cancer portions). The ratio of black to white pixels turns out to be 10000:1. In order to tackle this problem we introduce a two-step segmentation scheme shown in Fig. 2.3. This scheme comprises of two blocks

1. **CNN Classifier**. This module comprises a convolutional neural network (CNN) [19] classifier which is responsible for determing whether a 2D CBCT

slice has cancer portion or not. Depending on the output y, it is forwarded to the segmentation block or a null image is returned. This helps in relieving the class imbalance to some extent. VGGNet [20] or AlexNet [19] has been used to implement this module.

2. **Segmentation Block**. This module is used only when the output from the former module $y = 1$. This block consists a pixel-wise segmentation network which in our case is U-Net [3]. The same setup of U-Net is used with modified loss functions as discussed in Section 2.2.

The output of the two-step segmentation scheme is shown in Fig. 2.4. It is evident that the network is still unable to differentiate between soft tissues and cancer patch.



(a) (b)

Figure 2.4: (a) Original scan (b) Predicted output

Region Proposal Networks

The results of the U-Net architecture even after using different loss functions were not impressive. U-Net was unable to capture the difference between the cancer patches and the soft tissue cells as both of them were having exactly same hue range. In this approach we aim to figure out regions where the probability of cancer is higher. Region proposal networks [21] come into play as they output bounding boxes to recognize the object required to be segmented. There are several variations of RCNN [21] like Fast-RCNN [22] and Faster-RCNN [23] which

provide state-of-the-art performance on competitive datasets. In our task we have used Single Shot Detector [16], as it requires less data points and computation by eliminating proposal generation, feature resampling stages and processing them in a single network.

Single Shot Multibox Detector (SSD)

The Single Shot Detection [16] approach is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections. SSD [16] only needs an input image and ground truth boxes for each object during training. The SSD architecture has two major parts (a) Base Network (Standard Image classification architecture) (b) Auxilary Network.

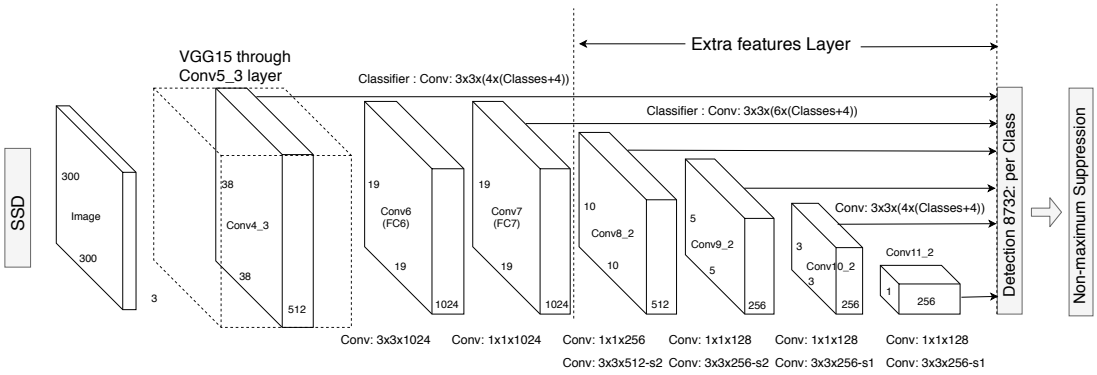


Figure 2.5: Single Shot Detector Network Architecture

Network Architecture

The Base Network used in the SSD architecture is a benchmark image classification architecture (eg. VGG16 [20], VGG32 [20], GoogleNet [24] etc.). In our experiments we have deployed the VGG16 [20] network architecture as the base network. In the original work, the base networks were prepared utilizing expansive scale datasets like ILSVRC which has a large number of data points. In our work,

the quantity of samples so as to train the base network is constrained and along these lines preparing the base network legitimately is a testing assignment.

Multi-scale feature maps: Convolutional feature layers are added to the end of the truncated base network. These layers decrease in size gradually and allow predictions of detections to occur at different scales. The convolutional model for predicting detections is different for each feature layer (Overfeat [25] and YOLO [26] that operate on a single scale feature map).

Detection using Convolutional predictors: Added feature layers using a set of convolutional filters produce a constant set of detection probabilities. These are illustrated on the right side of the SSD network architecture in Fig. 2.5. A feature layer of size $m \times n \times p$, the fundamental element for predicting parameters of a potential detection is a $3 \times 3 \times p$ kernel that creates either a score for a class, or a shape offset relative to the default box coordinates. The bounding box offset output values are measured relative to a default box position relative to each feature map location.

Default boxes and aspect ratios: The default boxes tile the feature map in a convolutional manner, so that the position of each box relative to its corresponding cell doesn't change. At each feature map cell, we predict the offsets relative to the default box shapes in the cell, as well as the per-class scores that indicate the presence of a class instance in the boxes. Specifically, for each box out of k at a given location, we compute c class scores and the 4 offsets relative to the original default box shape. This results in a total of $(c+4)k$ filters that are applied around each location in the feature map, yielding $(c+4)kmn$ outputs for a $m \times n$ feature map. The default boxes are similar to the anchor boxes used in Faster R-CNN [23], however we apply them to several feature maps of different resolutions. Allowing different default box shapes in several feature maps let us efficiently discretize the space of possible output box shapes.

Performance on CBCT Data

The SSD was trained on the 2D CBCT slices to validate the performance, the output mask was fed after forming a bounding box around the cancer patch. In order to train the base network, we used 2D samples of CBCT scans for differentiating between cancerous and non-cancerous slice. We trained the network using our dataset, where the number of non-cancerous slices was much more compared to the number of cancerous slices. The base network failed to achieve acceptable classification accuracy. Upon visualization it had been observed that the features being learnt were arbitrary and did not correspond to the cancer patch.

Chapter 3

Proposed Model

We have studied the performance of various deep learning techniques in the previous chapter. The key problem we were facing during segmentation is the similarity in the pixel intensity of soft tissues and cancer patch. Larger convolution filters were unable to localize the cancer and detected any non-zero pixel as cancerous, and vice-versa. To tackle this problem we develop a two-step solution where we get rid of the soft tissue region and focus on the interior region of the lungs. Thresholding techniques are not applicable to perform this operation, as Hounsfield Unit [27] is not benchmarked in CBCT domain. In this chapter, we introduce two disparate models as already discussed, one responsible for (a) localization of cancer region and other is used for (b) pixel-wise segmentation. In the subsequent sections we will discuss the two modules in detail.

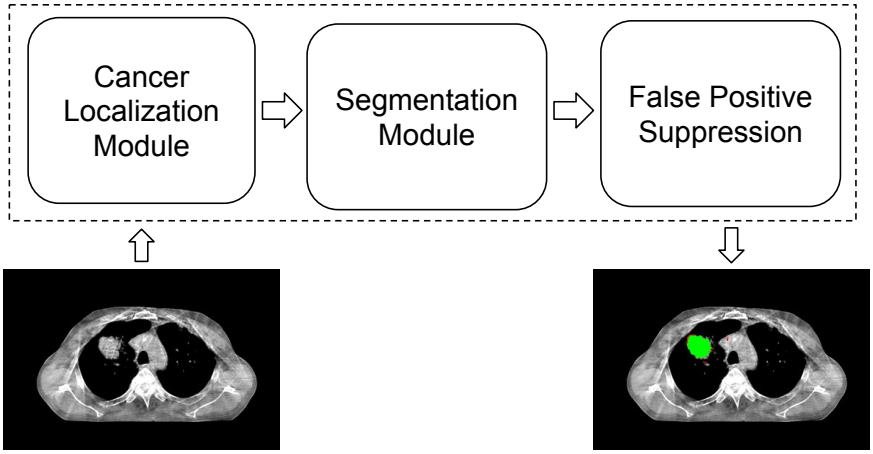


Figure 3.1: Schematic overview of the proposed model

Cancer Localization Module

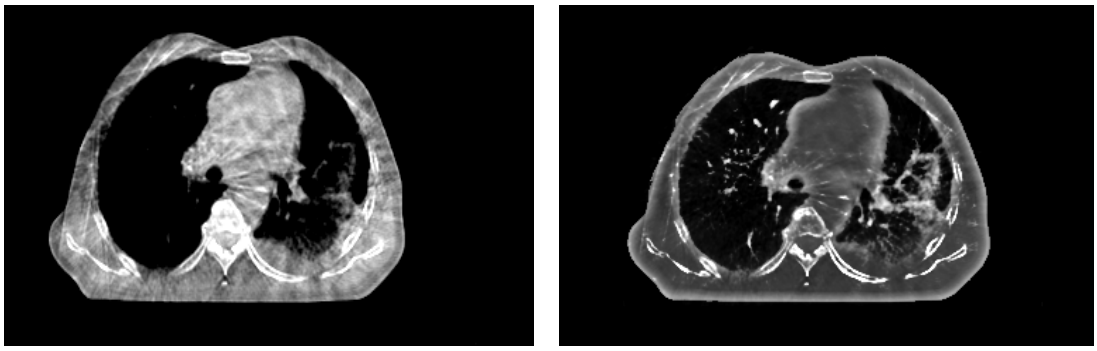
In this section we describe the cancer localization technique using a set of image processing techniques. We used image enhancement techniques and other methods which leverages the structural characteristics of the image in order to localize the cancer patch region. The complete algorithm for the localization module is described in Appendix A and B. The individual units for the cancer localization is mentioned in the following sections.

Local Rank Transform (LRT)

The challenge in segmentation of lung cancer in CBCT images is that the pixel intensity range of cancer and soft tissues are similar. To tackle this problem, we use image enhancement using local rank transform for the 3D CBCT images. LRT [28] computes the local rank of the pixel under consideration in a defined neighborhood. A variant, adaptive δ-LRT, is defined as the number of pixels less than it in the neighborhood by a margin of at least δ. The neighborhood in our experiments is considered as a 3D voxel of dimensions $(7 \times 7 \times 7)$. For the purpose of image enhancement [28], two adaptive thresholds δ_1 and δ_2 are used, and the final enhanced image is computed as

$$\hat{\mathbf{X}} = X + \lambda_1 \mathrm{LRT}_{\delta_1}(X) + \lambda_2 \mathrm{LRT}_{\delta_2}(X)$$

where, X is the original image, $\hat{\mathbf{X}}$ is the enhanced image, $\delta_1 = 256$, $\delta_2 = -40$, $\lambda_1 = 0.95$ and $\lambda_2 = 0.05$. It can be shown that positive δ value correspond to the edges in the image while the negative δ extracts the smoother portions in it. A linear combination of these along with the original image is helpful in noise suppression. Figure 3.2 shows a sample original image and its corresponding local rank enhanced image.

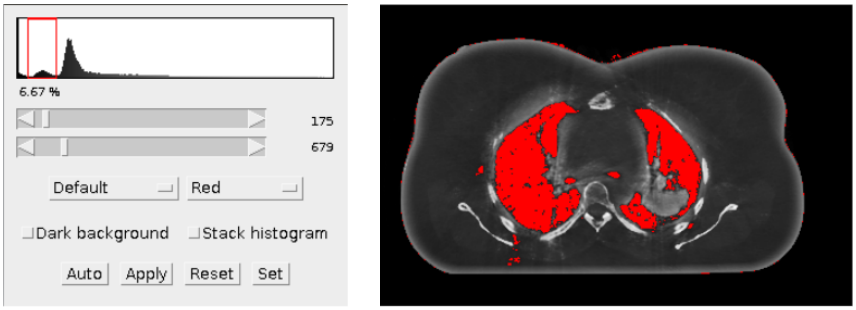


(a) (b)

Figure 3.2: (a) Original and (b) LRT enhanced CBCT scan

Thresholding Stage I

From the characteristics of the LRT output image it is observed that the histogram of the image possess a particular characteristic. The histogram shows that the pixel intensities in the image are concentrated around two regions (shown in Figure 3.3), thresholding around the first region gives us the internal lung region where the cancer resides. By experimentation over several images, the threshold range was adjusted to [100, 700] for uint16 image. The thresholding was performed individually on 2D slices of CBCT scan. The region marked in red shows the output of this thresholding process.



(a) (b)

Figure 3.3: (a) Histogram and (b) Output for Thresholding Stage I

Connected Component Retrieval

In order to extract the complete inner portion of lung in the scan, we need to form a mask of similar shape over the image. We use the thresholded output of the previous step and retrieve the connected components [29] where the number of elements are above a certain threshold. The threshold is determined empirically and it is necessary to get rid of minor noises from the thresholding stage. An example of connected components retrieved is shown in Figure 3.4a. To form a homogeneous mask, we use binary filling and binary closing to the output to obtain Figure 3.4b.

(a) (b)

Figure 3.4: (a) Populated Connected Components (b) Binary Filling and Closing Output

Convex Hull

The convex hull of a set of points S in n dimensions is the intersection of all convex sets containing S. For N points $\{p_1, \ldots, p_N\}$, the convex hull C is then given by the expression

$$C = \{\sum_{j=1}^{N} \lambda_j p_j : \lambda_j \geq 0 \; \forall \; \sum_{j=1}^{N} = 1\}$$

In our binary mask we apply convex hull formation algorithm (Graham Scan [30]) in order to obtain a boundary over the mask pertaining to the entire internal region of the lung. Graham scan [30] is an efficient algorithm for calculating the convex hull in $O(n \log n)$ time, where n is the number of points. We obtain a convex hull

of the mask obtained in the last step and apply it on the original image to obtain a region as shown in Figure 3.5.

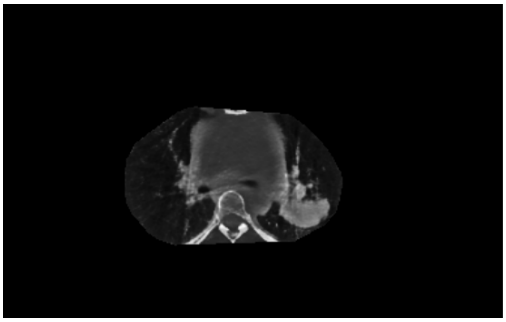
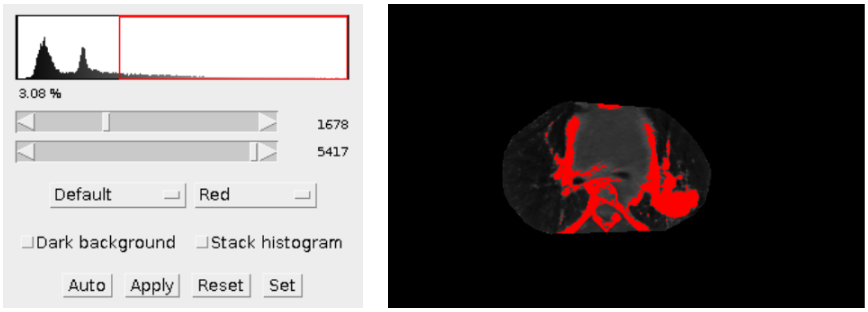


Figure 3.5: Convex Hull output after retrieval of connected components

Thresholding Stage II

The histogram of the output image is obtained and second step of thresholding is performed to obtain an attention map of the cancer. Emperically the thresholding is chosen for pixel intensities more than 1700 (in Figure 3.6 it is 1678). The histogram and attention map formed after thresholding is shown in Figure 3.6.



(a) (b)

Figure 3.6: (a) Histogram of the output (b) Attention map after thresholding

Cancer Extraction using Structural Characteristics

In this section we use a variation of the methodology discussed in last section in order to locate the cancer patch with more precision. We observe that the connected components obtained have a missing one portion which is present in the other lung portion. Our algorithm tries to capture that portion as it is most

likely to be the cancer patch. This helps us to get rid of the soft tissues in the central portion thereby reducing false positive rate. The subsequent sections discuss the steps followed to obtain the output mask.

Lung Patch Separation

This involves division of the image into two by forming a line which separates the two lungs through the middle. The line is obtained by the taking into account the centroid of the nearest (lowest x-value) and furthest (highest x-value) connected component. A line is formed at the middle of the two x-values obtained, which separates the image into two. Samples of the images formed are shown in Figure 3.7. The output of this step is used to localize the cancer region using the following methods.

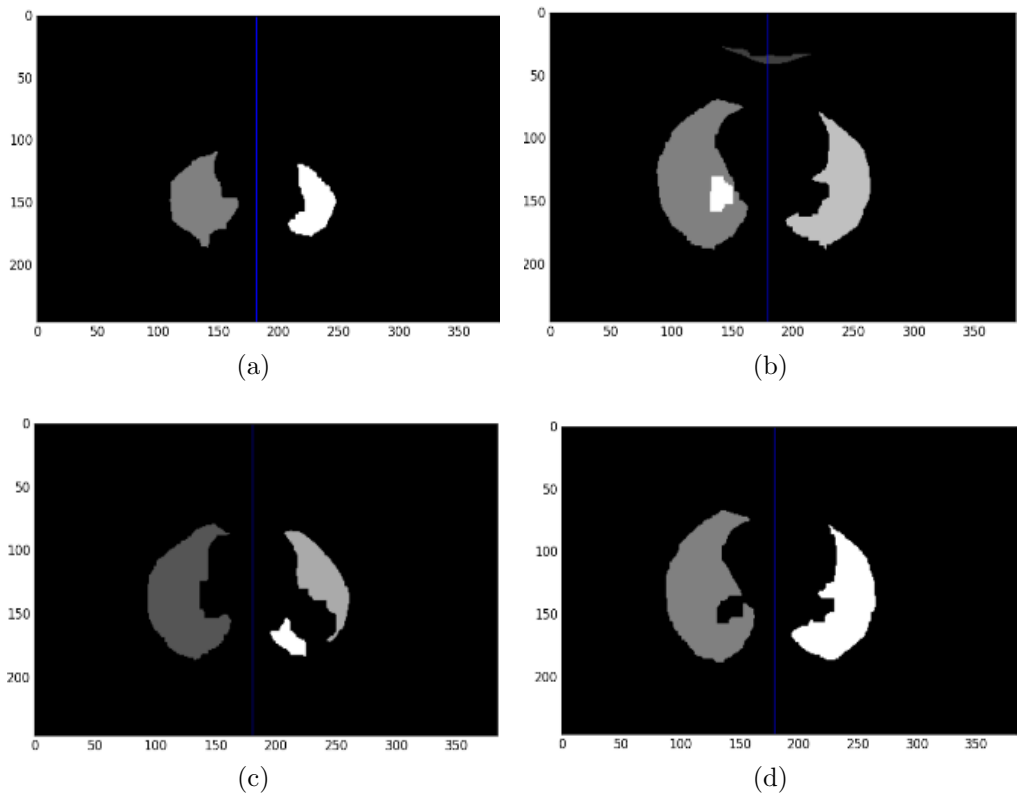


Figure 3.7: Sample of the central lines formed for different lung patches. The line of division is represented in blue.

Template Matching

It is observed from samples in Figure 3.7, the hollow portion in one of the lung patch presumably belongs to the cancer. In order to efficiently extract this region we perform template matching between the two lung patches. Template matching [31] wherein a template t is matched for maximum correlation in two-dimensional image f.

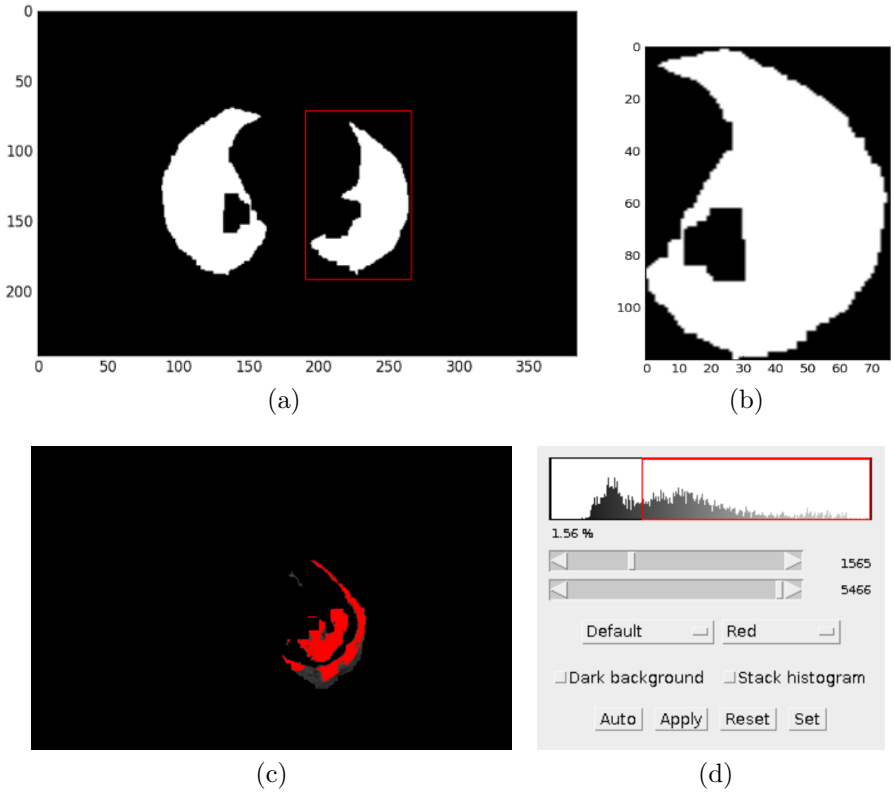


Figure 3.8: Sample of the central lines formed for different lung patches. The line of division is represented in blue. (a) Histogram and (b) Output after thresholding in the reconstructed image from the convex hull outputs.

Let $f(x, y)$ denote the intensity value of the image f of the size $M_x \times M_y$ at the point (x, y), $x \in \{0, \ldots, M_{x-1}\}$ $y \in \{0, \ldots, M_{y-1}\}$. The pattern is a template t of the size $N_x \times N_y$. In order to calculate (u_{pos}, v_{pos}) of the pattern in the image f we computer the Normalized Cross Correlation (NCC) [31] value (γ) at each point (u, v) for f and the template t has been shifted by u steps in the x-direction and v steps in y-direction. The value of γ is calculated as follows

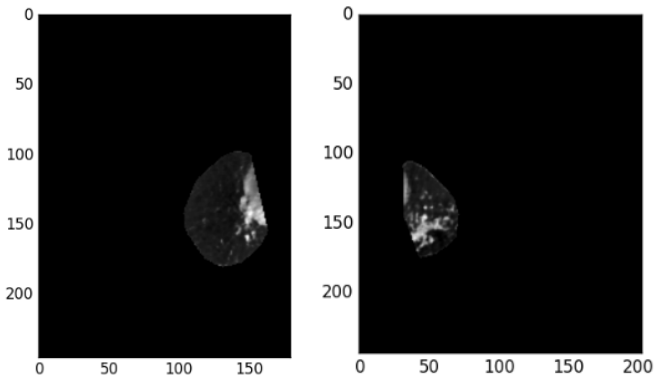
$$\gamma = \frac{\sum_{x,y}(f(x,y) - \hat{f}_{u,v})(t(x-u,y-v) - \hat{t})}{\sqrt{\sum_{x,y}(f(x,y) - \hat{f}_{u,v})^2 \sum_{x,y}(t(x-u,y-v) - \hat{t})}}$$

In our problem, we divide the image into two halves and take the largest connected component as the template. Using this template we search in the other half of the image. The sample output of the above algorithm is shown in Figure 3.8a. The bounding box denotes the region of maximum correlation.

After template matching [31] we perform image subtraction between the template and the maximum NCC region as shown in Figure 3.8c to obtain the cancer patch. The method described is only applicable for detecting cancer patches which are significantly large in size.

Cancer Extraction using Structural Characteristics II

In this section, we describe a variant which even detects the cancer which are quite small in size but at the same time leads to higher false positive rate. We try to do away with the soft tissues in the central portion of the lung region and focus on only the lung region. This method is also computationally cheap and doesn't require calculation of NCC for template matching as in the previous section.



(a) (b)

Figure 3.9: (a) Left half of the 2D slice (b) (a) Right half of the 2D slice of convex hull outputs

Convex Hull on Independent Lung Patches

Two dimensional CBCT slices are divided into two halves containing the lung patches. We apply Graham scan to find out the convex hull in each the left and right half sections of the image. All the points within each of the two hulls were used as a mask independently and applied on the input image. The shape of the masks after splitting are shown in Figure 3.9a and 3.9b. The two halves are concatenated to form a single image of the same dimension as the original scan shown in Figure 3.10.

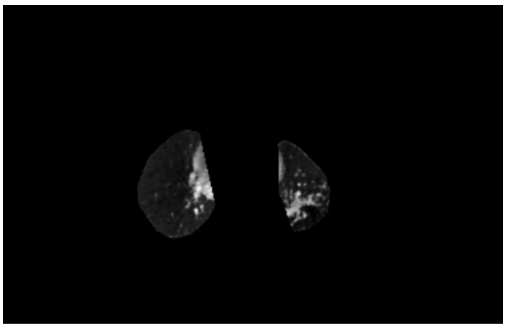
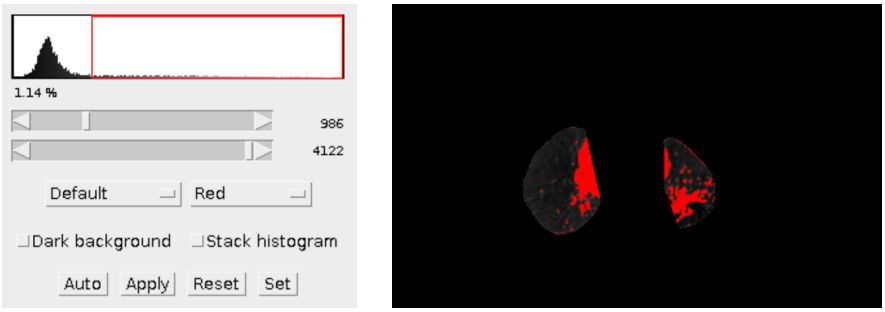


Figure 3.10: Concatenated image formed from the convex hull outputs

The output is then thresholded to form the attention on the probable cancerous region as shown in Figure 3.11. As evident from the histogram[1], the pixel intensities are situated around a certain central pixel value. The intensity value from the right tail-end of the intensity block shown in Figure 3.11a.



(a) (b)

Figure 3.11: (a) Histogram and (b) Output after thresholding in the reconstructed image from the convex hull outputs.

[1]All visualization was performed using open sourced software Fiji-ImageJ

Pixel-wise Segmentation techniques

In this section we will tackle the problem of segmenting individual pixels from the 3D CBCT images. These techniques are applied after the localization of cancer region is done. In our experiments, the two modules work independent of each other. The setup of our experiment included a predefined 3D voxel around the cancer region, the pixels in the voxel were considered only. For every pixel we have extracted a feature vector corresponding to it which is used for classifying it into a cancerous or non-cancerous pixel. For classification we have used k-NN, Random Forest Classifier and Multi-layered Perceptron. k-NN is time consuming and memory exhaustive, whereas Random Forests is much faster and therefore is preferred. In order to make the processing time lower for k-NN, a KDTree approach was incorporated to retrieve neighbors faster. Feature selection is also an important factor in determining the performance of the overall segmentation procedure. The feature selection techniques are described in the following sections.

Cubic Voxel

CBCT images were converted into multiple feature vectors in the shape of 3 $\times$ 3 $\times$ 3 voxels surrounding a central pixel forming an 1D feature vector of length 27. The 3D representation of the voxel is shown in Figure 3.12.

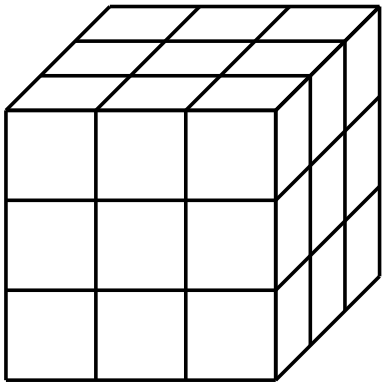


Figure 3.12: Representation of the 3D voxel feature vector. Each individual 3D cube denotes a specific pixel in space.

Histogram of Oriented Gradients (HoG)

HoG [32] is a global image feature descriptor used mainly for object detection. HoG computes the occurrence of similar gradient orientations in localized image subsections. This is helpful in detecting the image shape and edge directions. HoG although a global descriptor has been used as a local descriptor in our experiments. Two dimensional CBCT slices are divided into 16×16 subsections which are used as feature input to HoG for classification of the central pixel. The length of the resultant feature vector formed is 16.

Sparsified Voxel

Cubic voxel considers all pixel intensities in a specified voxel. The voxel dimension is small and hence does not always capture the larger scenario. The voxel dimensions cannot be made very large as the feature vector length increases leading to increasing computational complexity. To overcome this difficulty, we use variable voxel sizes and consider only corner and face pixel intensities in the voxel shown in Figure 3.13.

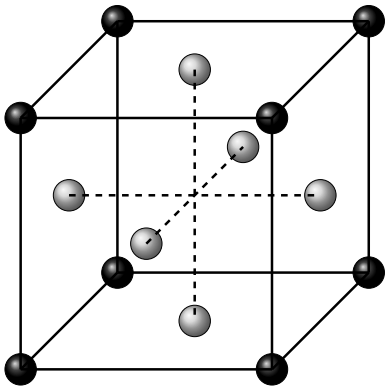


Figure 3.13: Representation of sparsified voxel feature vector. The black pixels denote those situated at the corners and gray pixels denote those at the face of each side.

In the experimentation section we describe the performance, time complexity and efficacy of the discussed approaches. We also try different parameters and compare our methods with the conventional learning approaches.

False Positive Suppression

The segmentation map formed after segmentation process is quite noisy due to the various noise present in CBCT scans. This leads to increase in false positive as shown in Fig. 3.14a. The noise formed after segmentation appears as salt and pepper noise. We can ignore these output pixels using the prior knowledge that cancer occurs as a cluster.

Median Filtering

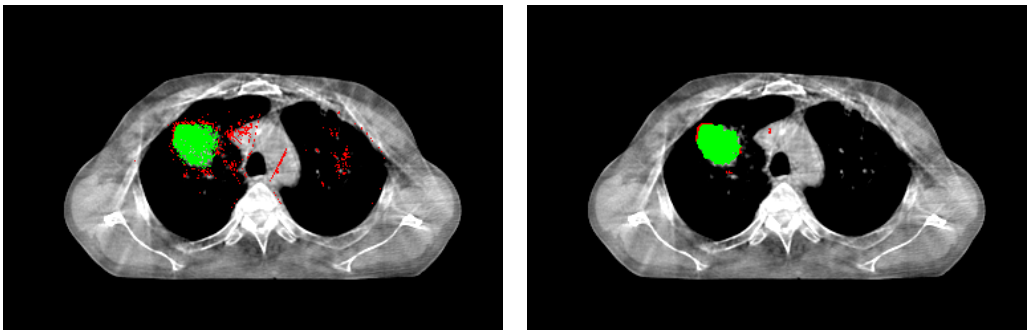
Median filtering aids in the elimination of small connected components which appear as noise in Fig. 3.14a. We use median filtering in our experiments as the computation is less exhaustive, output is shown in 3.14b. We see that this step reduces the false positive rate by a significant margin and the segmentation map is quite close to the groundtruth as shown in Fig. 3.14c. We discuss the detailed results in Section 4.5.

Hierarchical Clustering

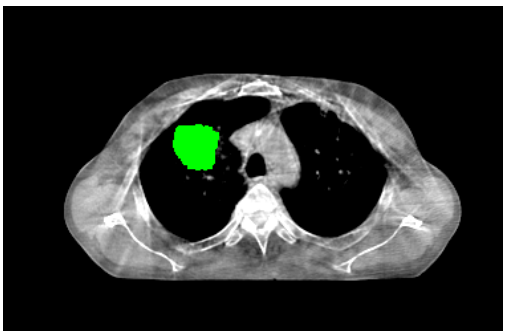
Median filtering suffers from several disadvantages as in our case it acts as an erosion operator and also removes certain true positive regions along with the false positive. In order to overcome this we perform hierarchical clustering of the connected components in the segmentation map. We have used a modified version of the Hausdorff distance where we consider the minimum (instead of maximum) distance between any two points belonging to two different sets X and Y

$$d(X,Y) = \min\{\sup_{x\in X}\inf_{y\in Y} d(x,y), \sup_{y\in Y}\inf_{x\in X} d(x,y)\}$$

The algorithm proceeds by considering only connected components whose volume is large and calculates the distance between all other connected components. If the distance is below a certain threshold, the connected components are merged to



(a) (b)



(c)

Figure 3.14: (a) Saliency map formed after segmentation (b) Saliency map after segmentation followed by median filtering (c) Groundtruth of the original scan. Pixels labeled by green denotes the true positive cancer pixels while the pixels labeled by red illustrate the false positive regions.

form one. In the next iteration, the distance to the merged connected component is computed and any other component within the threshold is taken into account. This continues untill there is no connected component with distance below the threshold. The complete algorithm is presented in Appendix C. This method is computationally much more expensive than median filtering. This technique although provides a higher true positive rate than median filtering, there is a compromise in the false positive rate which is slightly higher than filtering. The detailed results are discussed in Section 4.5.

Chapter 4

Results and Discussion

In this section we will describe the setup, evaluation metric and analyze the performance of the localization and segmentation modules described in the previous section. Our evaluation will mainly focus on the segmentation portion with many variations of voxel dimensions and shapes used.

Dataset

For experimental purposes, clinical data has been used for low-dosage 3D CBCT scans in which the cancer patches are labeled by radiation oncologists. All the patients had a visible cancer region and the scans were part of the radiation treatment process. The original scans were available as 3D DICOM images, which were reconstructed to '.tiff' images for our processing. Each 3D CBCT scan contains around 54/64 2D slices each. Our dataset had scans from 45 different patients with a scan every week over a span of 5 to 7 weeks, totaling to 307 volumes of 3D scans. When the entire data is converted into 2D slices it has 13502 image slices and their corresponding masks. Around 5000 image slices has a cancer patch within it. For cancer classification, we use the entire dataset but for cancer localization we only utilize the image slices which are labeled as cancerous. Every 3D CBCT scan had its corresponding binary 3D labeled output denoting its cancer region, which serves as the ground-truth for the segmentation module.

Setup

The training process for neural networks were performed using Nvidia Quadro M5000 8GB GDDR5 (2048 CUDA cores) GPU on a system with Intel Xeon E5, 2630 with 10 cores, 20 cores with Hyperthreading. The dataset is originally in '.dcm' format which is then converted to '.tif' for processing purposes. The output of the Local Rank Transform (LRT) is stored in a 'uint16' format which has been useful in image enhancement. The deep learning architectures were implemented using TensorFlow [33].

Results

In this section, we discuss the performance of both the deep learning techniques and proposed modules in localization and segmentation of cancer from CBCT scans. For all deep-learning based segmentation we have used U-Net as the segmentation block.

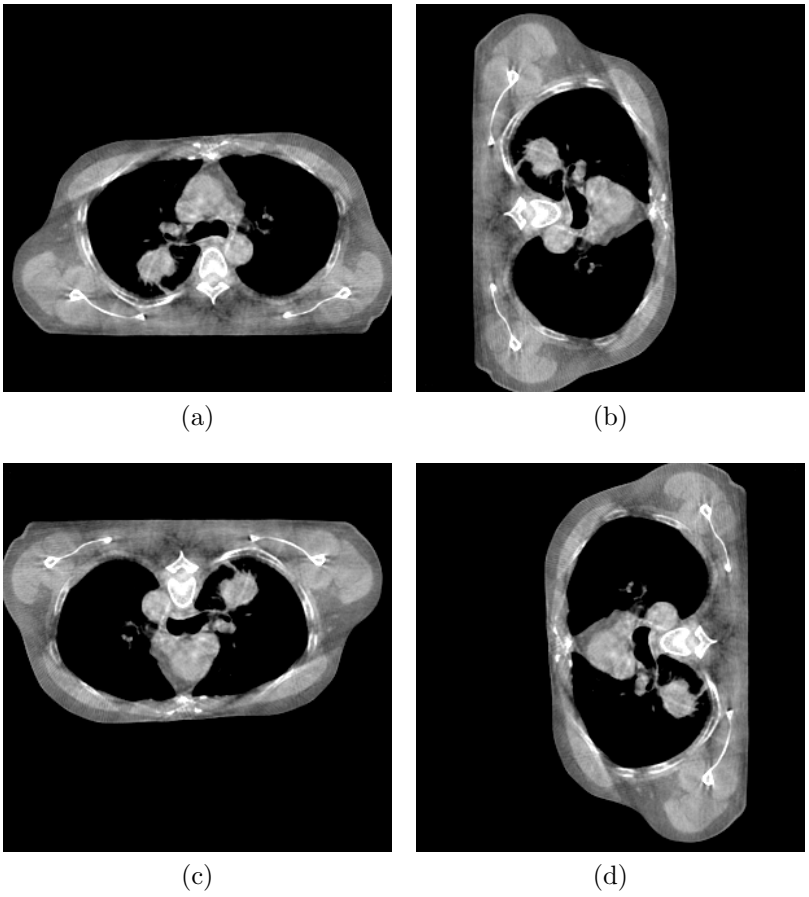
U-Net Performance

The U-Net architecture was applied on the CBCT data using different loss function in order to deal with the high-bias in the data. Before going into details of the performance of U-Net using individual loss functions, we will discuss few image augmentation techniques utilized to improve the strength of the dataset.

Image Augmentation techniques

Conventional image augmentation techniques involve random rotation, shifting, scaling, adding noise, lighting condition, perspective transform etc. In our case of CBCT scans, only rotation and forming smaller image patches is only relevant. We have deployed the two techniques as discussed below

(a) **Image Rotation**: The original 2D slice from a CBCT image is rotated by 90°, 180°and 270°increase the size of dataset by 4 times as shown in Fig. 4.1.



(a) (b) (c) (d)

Figure 4.1: (a) Original scan is rotated by (b) 90°(c) 180°and (d) 270°

(b) **Smaller Image patch**: This process involves taking small 2D patches around the cancer region of size 64×64, which are then fed to the segmentation block (U-Net).

Cross Entropy Loss

The entire dataset without any preprocessing was fed into the U-Net [4] architecture for segmentation. The null data points created a high-bias in the dataset. Even in the labeled data points the ratio of the white to black pixels were 1:400. The overall dataset has a black to white pixel ratio of 1:10000. This high-bias in the data caused the learning process to convert all the pixels to black as it would still achieve a low loss even it misses the tumor region. The performance of the

two variants of the localization module are mentioned in detail below

Verification Accuracy = 73%

Training Loss = 0.46

Dice Coefficient Loss

Using dice-coefficient loss the model seemed to fit the training data better. Fig. 4.2 illustrates the training and verification loss on our CBCT dataset.

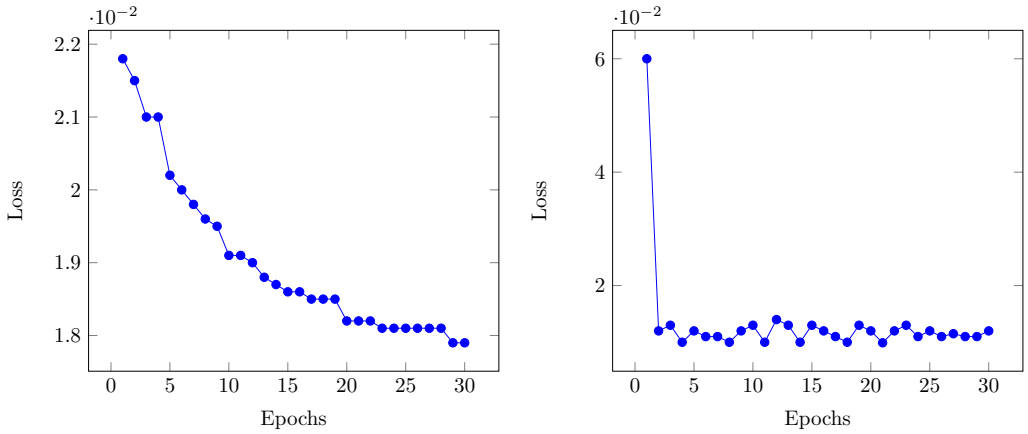


Figure 4.2: (a) Training and (b) Verification loss plots using dice-coefficient loss function

Weighted Cross Entropy Loss (WCE)

Weighted Cross-Entropy Loss function also failed to produce a suitable distinction between cancerous and non-cancerous patches. The visualization of the output using weighted cross-entropy is shown in Fig. 4.3.

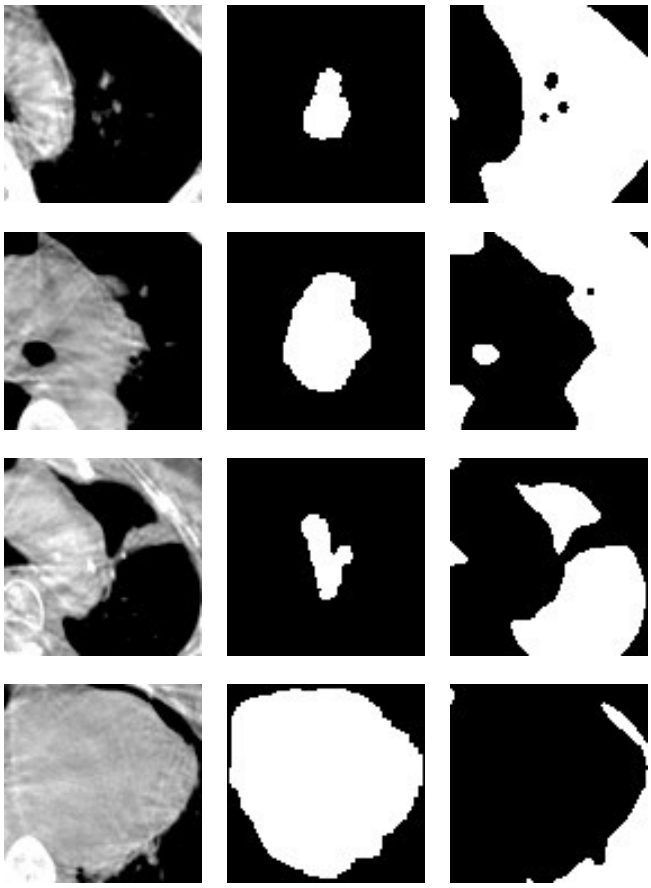


Figure 4.3: Output from U-Net architecture when small patchs of CBCT scans were used as input. Loss function used was weighted cross entropy function.

Performance of Proposed Model

Localization Module

(a) **Performance of Template Matching**: This methodology was applied to a large number of image slices to study its characteristics. Our method was able to recognize large cancer masses, on the other hand it completely ignored when the cancer size was small. Our metric for measurement was set to be "The percentage of the original cancer captured by the model output" (True Positive). This approach is able to detect only large cancer patches and doesn't detect smaller patches partially. For large cancer masses accuracy: $\geq 50\%$ (We define large cancer masses as those for which this method is at least able to identify a single pixel

of the cancer region).

(b) **Performance of Convex Hull on Lung Patches**: Unlike the previous method which is instrumental in detecting large cancer masses. This is a more robust method because it always captures some part of the cancer irrespective of its size. True Positive rates are higher using this method. When tested across all 2D CBCT slices, this method was able to achieve a 99:1 hit ratio, i.e. in only 1 out of 100 samples it missed out the cancer lesion completely.

Segmentation Module

We will now discuss the efficacy of this module under the various setups discussed in the previous section. For the segmentation experiments we have prepared a miniature dataset representing only the feature space. For a given image, a bounding box is formed of shape ($32 \times 64 \times 64$) around the centroid of the cancer globule.

(a) **Cubic Voxel**. For cubic voxels as already discussed the length of the feature vector is 27. Pixels within the bounding box are considered for feature formulation. The total size of the dataset formed is: 33067008 (33M). In order to determine the hyperparameter k for kNN classification we ran simulations with only 50000 samples. In our experiments, we define accuracy as $\text{Accuracy} = \dfrac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$

k-value	Accuracy (%)	Time (sec.)
1	73.79	3211.49
2	75.38	3710.57
3	73.82	3825.09
4	74.81	4088.90
5	74.49	3795.24
6	73.76	3864.43

Table 4.1: Performance with varying k-value

The above set of experiments were evaluated on 20% of the dataset samples from unseen patient during training. From Table 4.1 for $k = 2$ we get the best performance, but as k should ideally be odd we settle for the next best performance at

$k = 5$ for the following experiments.

This experiment was performed with 5M data samples and tested on 30% of the dataset from unseen patients.

k=5, **Accuracy**: 73.92% **F1 Score**: 0.6006

In order to analyse the effect of filtering, we have also tried a variant in which median filtering is performed as a preprocessing step before feature selection and classification. kNN classification was used with $k = 5$.

Accuracy: 75.29% **F1 Score**: 0.5812

Precision: 0.5779 **Recall**: 0.5858 **Time Taken**: 10775.28s

From this experiment, we observe that the filtering has an adverse effect on the segmentation process and thereby we can infer that raw pixel intensities contain much more relevant information.

(b) **Histogram of Oriented Gradients (HoG)**. HoG [32] is a global feature descriptor, but for our setup we need features in a localized region around a pixel. Therefore, we use a 16×16 2D slice as the input to HoG, the output is 1D vector of length 16. In these set of experiments, we randomly sample 5000 positive and negatively labeled pixels for a single scan, the total dataset size is 2M. The learnt model was tested on 20% of the data from unseen patients. Taking these feature vectors into account, we perform classification using k-nearest neighbour, MLP and Random Forests Classifier. The performance of the classification techniques are shown in Table 4.2.

Classifier	Accuracy (%)	F1 Score	Precision	Recall	Time
Nearest Neighbor	57.55	0.5747	0.5761	0.5755	2871.97
Random Forests	59.05	0.5841	0.5965	0.5905	242.11
MLP	**62.38**	**0.6220**	**0.6262**	**0.6238**	38.17

Table 4.2: Performance with varying voxel dimensions for HoG feature descriptor using kNN, MLP and Random Forests Classifier.

The results show that the performance is not affected much and is similar to the Cubic voxel feature descriptor. The computation time is reduced by a large margin

as the feature length is much less. Random Forest classifier on the other hand has the advantage of reducing the computation time by 10 folds, while delivering similar performance.

Thereby, we see that the HoG achieves similar performance as the cubic voxel feature descriptor but in much less time and using lesser memory. However, extracting the features from the image is computationally expensive.

(c) **Sparsified Cuboid Voxel**. In this feature descriptor, we consider only certain pixel values in a cuboid. Fig. 3.13 shows the positions in the voxels being considered, 14 pixels values are considered in a voxel. Another value is concatenated with this feature vector which is the LRT of the central pixel value in the cuboid. The intuition behind adding this value is that we have noticed that the cancer globule is the one whose shape changes most rapidly along the z-axis. In order to capture this feature and distinguish it from soft tissue region LRT value is added. The dataset sampling and train-test division in these experiments were identical to those mentioned for HoG setup. The feature length in this setup is 15. The performance on varying cuboid sizes are described in Table 4.3.

Voxel dim.	**Classifier**	**Accuracy (%)**	**F1 Score**	**Precision**	**Recall**	**Time (sec.)**
(5×5×5)	k-NN	72.56	0.7255	0.7257	0.7256	1200.66
	RF	72.27	0.7223	0.7242	0.7228	92.22
	MLP	**73.69**	**0.7339**	**0.7482**	**0.7369**	155.44
(5×5×9)	k-NN	72.95	0.7295	0.7295	0.7295	1131.03
	RF	72.88	0.7283	0.7306	0.7288	92.83
	MLP	73.12	0.7297	0.7367	0.7312	170.19
(5×5×11)	k-NN	72.41	0.7241	0.7241	0.7241	1163.85
	RF	72.89	0.7285	0.7303	0.7289	121.53
	MLP	73.52	0.7327	0.7442	0.7352	93.19
(7×7×11)	k-NN	71.32	0.7132	0.7133	0.7132	1172.90
	RF	71.79	0.7174	0.7194	0.7179	108.45
	MLP	71.89	0.7150	0.7314	0.7188	**67.88**

Table 4.3: Performance with varying voxel dimensions for Sparsified Cuboid feature descriptor using k-Nearest Neighbour (k-NN), Multi-Layered Perceptron (MLP) and Random Forests (RF) Classifier.

It is seen that the incorporation of LRT doesn't impact the performance by much or at all, however changing the z-dimension of the voxel shows performance enhancement. Voxel dimension (5×5×5) using MLP classifier shows the best performance in terms of accuracy, F1 Score and Recall. However classification techniques using Random Forests deliver almost similar performance in approximately 10-fold reduced computation time.

Table 4.4 summarizes the performance of the three discussed segmentation techniques. It is evident that the sparsified cuboid performs better in terms of both performance and computation time by a significant margin. In the subsequent sections, this setup has been used for generating the segmentation map.

Feature Descriptor	**F1 Score**	**Precision**	**Recall**	**Time (sec.)**
Cubic	0.5812	0.5779	0.5858	10775.28
HoG	0.6220	0.6262	0.6238	**38.17**
Sparsified Cuboid	**0.7339**	**0.7482**	**0.7369**	155.55

Table 4.4: Performance of the three segmentation techniques in their best setup.

Visualization

In this section we analyse the performance of our proposed model by visualizing the results of the end-to-end system localization followed by segmentation module. The prediction for 3 different CBCT slices are shown in Fig. 4.4. We see significant improvement in False negative reduction compared to Fig. 2.4 and Fig. 4.3. We also see some interesting features in the segmentation map.

In Fig. 4.4c, the cancer lesion lies on the chest wall which has soft tissues with the exact same intensity values making it difficult to segment. [8] used prior-knowledge from PET scans with random walks to tackle this problem. Our method efficiently deals with the same situation without using any external knowledge or focus on this corner case during the algorithm formulation.

We illustrate another use case of the segmentation procedure in Fig. 4.5 in order to prove the efficacy of our proposed model. Fig. 4.5a & 4.5b show the groundtruth

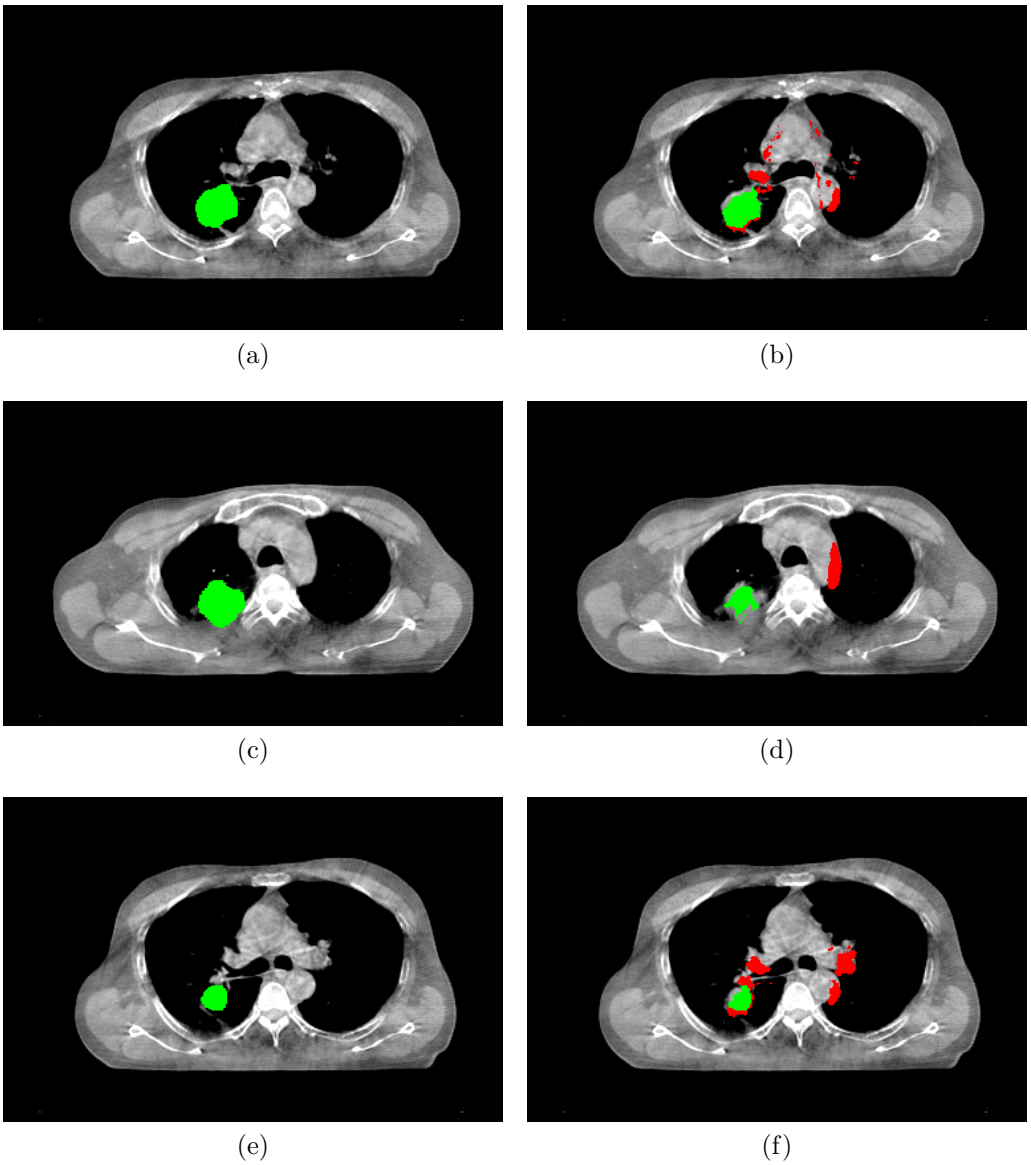
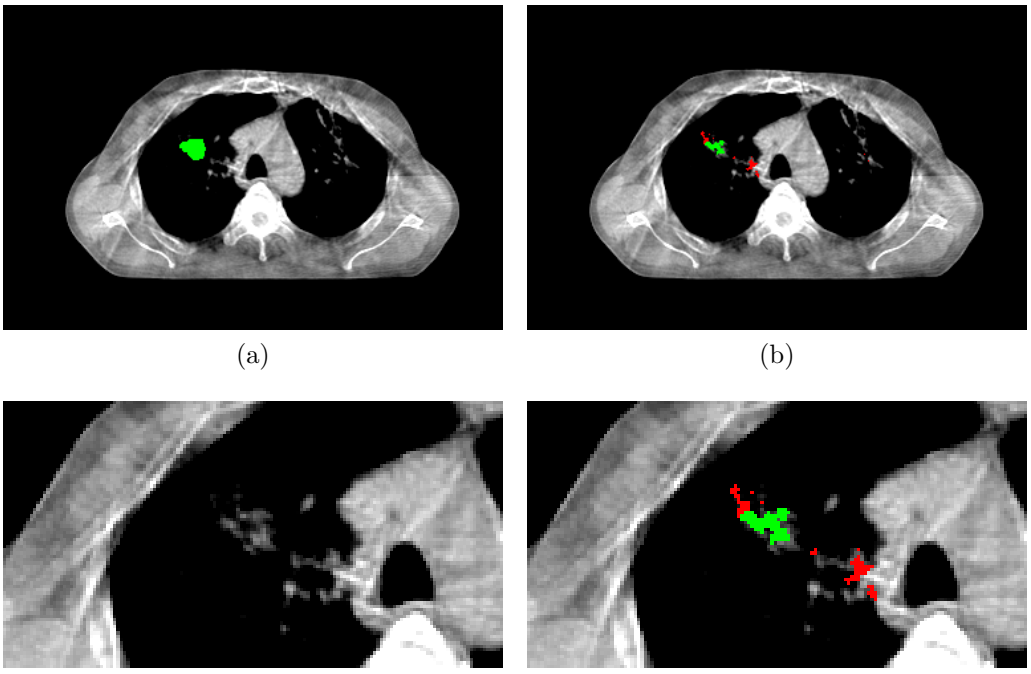


Figure 4.4: Grountruth and their corresponding predictions from the proposed model is shown. (a), (c) and (e) are groundtruth with green pixels being the segmented cancer region. (b), (d) and (f) represent the prediction with green pixels being the true positive and pixel labeled as red being the false positive output.

and segmentation map of a CBCT slice. Fig. 4.5c shows a magnified version of the cancer in the scan slice, the scan is noisy so the entire cancer lesion is not visible as a single connected component. There even exists null (black) pixels within the cancer groundtruth of the slice. Our model correctly labels them as cancerous even though the region is surrounded by null pixels which denote vacant space within the lungs. This particular feature is not possible using neural network

based architectures where the focus is mainly on the hue intensity of the pixel being classified.



(a) (b)

(c) (d)

Figure 4.5: (a) Groundtruth of a CBCT slice (b) Predicted segmented map of the slice (c) Magnified view of the cancer region in groundtruth and (d) predicted output.

False Positive Suppression

In this section we discuss the performance of the post-processing techniques performed towards false positive suppression in Section 3.3. Table 4.5 illustrates in detail the performance of the vanilla segmentation and two proposed method. As discussed using hierarchical clustering and median filtering, the false positive rate decreases along with an improvement in accuracy. Median filter causes an extra dip in true positive rate due to its erosion property. Hierarchical clustering maintains a significant true positive rate accompanied by a dip in false positive rate. True negative rate is also higher using median filtering which labels many pixels as non-cancerous due to its erosion property. Heirarchical clustering should be prefered if sensitivity is our concern, for high specificity median filtering should

be considered.

Post Processing	**True Positive**	**True Negative**	**False Positive**	**False Negative**	**Accuracy** (%)
Vanilla	**22.49**%	98.89%	1.11%	77.51%	98.34%
Median Filtering	21.28%	**99.24**%	**0.76**%	**78.72**%	**98.68**%
Hierarchical Clustering	21.99%	99.00%	0.98%	78.01%	98.44%

Table 4.5: Performance of the post-processing techniques after segmentation.

Latency Computation

Our proposed model consists of two disparate localization and segmentation modules each of which has computationally expensive units (eg. Connected component retrieval, Convex Hull formation etc). The pixel-wise segmentation takes the maximum amount of time around 2 hours, which can be brought down using simple heuristics like if all the pixels in a voxel are zero, then the prediction is zero. In order to reduce computation time, similar heuristics can be applied to other units as well. In heirarchical clustering, the distance is computed between the convex hull of two sets instead of computing the distance between every point in the sets. The details of computing time of each module is described in Table 4.6.

Module	**Unit**	**Time**
Localization	Convex Hull	80.64 s
	Connected Components	0.83 s
	Miscellaneous	60.08 s
Segmentation	Feature extraction	3.33 min
	Prediction	16.67 min
Post-Processing	Median Filtering	2.2 s
	Hierarchical Clustering	6 min
	Total	**28.36** min

Table 4.6: Latency computation for a entire 3D CBCT scan sample.

Chapter 5

Conclusion & Future Work

Conclusion

The project deals with an use-case of object recognition in medical domain. The task of lung tumor detection is challenging as it involves segmentation of intricate tumor region from soft tissue region. The experiments performed using the U-Net [3] architecture showed various drawbacks and positive sides. Processing of 3D images with low data points is a difficult task. Therefore, our method of selecting only 2D slices and detecting first whether it has tumor region reduces computational complexity. The process was carried out in steps involving U-Net [3] and other observations.

We had also approached the problem using region proposal networks, wherein instead of obtaining pixel-wise segmentation we train a SSD [16] for proposing regions according to their likelihood of containing a cancer patch. There were still issues in using this method as the base network wasn't able to learn the necessary features which are required for classifying whether a given slice has cancer patch or not. Transfer learning is not possible in our case as a large open dataset for CBCT scans does not exist, and transfer learning from CT domain to CBCT leads to errors due to domain discrepancy.

In this work, we have presented two novel techniques for localization and segmen-

tation of lung cancer regions from noisy 3D CBCT scans. A sequence of image processing techniques were deployed to extract the region of interest where the cancer region resides. Following the localization process, segmentation is performed around a specific region around the tumor region. We have shown experiments with a variety of feature descriptors and also introduced a new descriptor for the classification process. The classification is performed using conventional learning techniques (k-nearest neighbour, Random Forests and multi-layered perceptron). Our approach does not depend manual initialization of any image portion. Segmentation accuracy obtained using our technique outperforms other contemporary techniques by a significant margin on CT and CBCT scan domain.

Future Work

One of the issues still remaining in our proposed model is the false positive removal using the two techniques mentioned in Section 3.3. Although we have discussed ways to get rid of stray noise, the soft tissues may also be present as a cluster having similar volume as the lung cancer lesion. Our discussed methods does not tackle such a situation so the system suffers from a low true positive rate. This is caused mainly due to the convex hull step in the localization module when the scan is noisy, and the entire lung regions is deformed.

The key contribution of our work is the proposal of a cancer localization technique which does not require any learning. This reduces the computation time and does not depend on the large scale availability of data. Datasets involving CBCT are scarce and it is quite expensive to collect labeled data from skilled radiologists.

Appendix A

Localization Algorithm 1

Algorithm 1 Localization Algorithm 1

Require: Input 2D slice $\mathcal{X}$, where cancer lesion is to be localized

procedure LOCALIZE($\mathcal{X}$)
  $h \leftarrow$ HISTOGRAM($\mathcal{X}$)
  $\theta_1, \theta_2 \leftarrow$ THRESHOLDLIMITS(h)
  $\mathcal{M} \leftarrow$ THRESHOLD($\mathcal{X}, \theta_1, \theta_2$)
  $\mathcal{L} \leftarrow$ CONNECTEDCOMPONENTLABEL($\mathcal{M}$)
  $\mathcal{C} \leftarrow$ CONNECTEDCOMPONENTS($\mathcal{M}$)
  for each $l \in \mathcal{L}$ **do**
    if $l.label.area \leq \theta$ **then** $l \leftarrow 0$
    end if
  end for
  $\mathcal{L} \leftarrow$ BINARYCLOSING($\mathcal{L}$)
  $\mathcal{L} \leftarrow$ BINARYFILLHOLES($\mathcal{L}$)
  $m_l \leftarrow 0, m_r \leftarrow \infty$
  for each $c \in \mathcal{L}$ **do**
    $x, y = c$.centroid
    $m_l \leftarrow \min(x, m_l)$, $m_r \leftarrow \max(x, m_r)$
  end for
  $c_x \leftarrow (m_l + m_r)/2$
  $\mathcal{M}_l \leftarrow \mathcal{M}[1 \dots c_x]$, $\mathcal{M}_r \leftarrow \mathcal{M}[(c_x + 1) \dots n]$, $a \leftarrow 0$
  for each $c \in \mathcal{C}$ **do**
    if $c.area \geq a$ **then**
      $a \leftarrow \max(a, c.area)$
      $x_l, x_r, y_l, y_r =$ BOUNDINGBOX(c)
      $t \leftarrow c$
    end if
  end for
  if $t \in \mathcal{M}_l$ **then**
    $z \leftarrow \mathcal{M}_l$
  else if $t \in \mathcal{M}_r$ **then**
    $z \leftarrow \mathcal{M}_r$
  end if
  $r \leftarrow$ MATCHTEMPLATE(z, t)
  $\mathcal{O} \leftarrow r - t$
  return $\mathcal{O}$
end procedure

Appendix B

Localization Algorithm 2

Algorithm 2 Localization Algorithm 2

Require: Input 2D slice $\mathcal{X}$, where cancer lesion is to be localized

procedure LOCALIZE($\mathcal{X}$)
    $h \leftarrow$ HISTOGRAM($\mathcal{X}$)
    $\theta_1, \theta_2 \leftarrow$ THRESHOLDLIMITS(h)
    $\mathcal{M} \leftarrow$ THRESHOLD($\mathcal{X}, \theta_1, \theta_2$)
    $\mathcal{L} \leftarrow$ CONNECTEDCOMPONENTLABEL($\mathcal{M}$)
    for each $l \in \mathcal{L}$ **do**
        if $l.label.area \leq \theta$ **then**
            $l \leftarrow 0$
        end if
    end for
    $\mathcal{L} \leftarrow$ BINARYCLOSING($\mathcal{L}$)
    $\mathcal{L} \leftarrow$ BINARYFILLHOLES($\mathcal{L}$)
    $m_l \leftarrow 0, m_r \leftarrow \infty$
    for each $c \in \mathcal{L}$ **do**
        $x, y = c$.centroid
        $m_l \leftarrow \min(x, m_l)$
        $m_r \leftarrow \max(x, m_r)$
    end for
    $c_x \leftarrow (m_l + m_r)/2$
    $\mathcal{M}_l \leftarrow \mathcal{M}[1 \dots c_x]$
    $\mathcal{M}_r \leftarrow \mathcal{M}[(c_x + 1) \dots n]$
    $\mathcal{M}_l \leftarrow$ CONVEXHULL($\mathcal{M}_l$)
    $\mathcal{M}_r \leftarrow$ CONVEXHULL($\mathcal{M}_r$)
    $\mathcal{O} \leftarrow$ CONCATENATE($\mathcal{M}_l, \mathcal{M}_r$)
    return $\mathcal{O}$
end procedure

Appendix C

Hierarchical Clustering Algorithm

Algorithm 3 Hierarchical Clustering Algorithm

Require: Input 3D image $\mathcal{X}$, whose cluster is to be computed

$\mathcal{C} \leftarrow$ ConnectedComponents($\mathcal{X}$)
$\mathcal{S} \leftarrow \{\phi\}$
for each $c \in \mathcal{C}$ **do**
    if c.area $\geq \theta$ **then** (area threshold for cluster selection)
        $\mathcal{S}$.insert(c)
        $\mathcal{C}$.remove(c)
    end if
end for
$\Delta \leftarrow \infty$
repeat
    for each $s \in \mathcal{S}$ **do**
        for each $c \in \mathcal{C}$ **do**
            $d \leftarrow$ Distance(c, s)
            $\Delta \leftarrow \min(\Delta, d)$
            if $d \leq \beta$ **then**
                $s \leftarrow s + c$
                $\mathcal{C}$.remove(c)
            end if
        end for
    end for
until $\Delta < \beta$ (a small positive number)

References

[1] W. C. Scarfe and A. G. Farman, "What is cone-beam ct and how does it work?," *Dental Clinics*, vol. 52, no. 4, pp. 707–730, 2008.

[2] J. Rexilius, H. K. Hahn, M. Schlüter, H. Bourquain, and H.-O. Peitgen, "Evaluation of accuracy in ms lesion volumetry using realistic lesion phantoms," *Academic radiology*, vol. 12, no. 1, pp. 17–24, 2005.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[4] J. Garayoa and P. Castro, "A study on image quality provided by a kilovoltage cone-beam computed tomography," *Journal of applied clinical medical physics*, vol. 14, no. 1, pp. 239–257, 2013.

[5] S. Basu, L. O. Hall, D. B. Goldgof, Y. Gu, V. Kumar, J. Choi, R. J. Gillies, and R. A. Gatenby, "Developing a classifier model for lung tumors in ct-scan images," in *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pp. 1306–1312, IEEE, 2011.

[6] Y. Gu, V. Kumar, L. O. Hall, D. B. Goldgof, C.-Y. Li, R. Korn, C. Bendtsen, E. R. Velazquez, A. Dekker, H. Aerts, *et al.*, "Automated delineation of lung tumors from ct images using a single click ensemble segmentation approach," *Pattern recognition*, vol. 46, no. 3, pp. 692–702, 2013.

[7] Y. Tan, L. H. Schwartz, and B. Zhao, "Segmentation of lung lesions on ct scans using watershed, active contours, and markov random field," *Medical physics*, vol. 40, no. 4, 2013.

[8] H. Cui, X. Wang, M. Fulham, and D. D. Feng, "Prior knowledge enhanced random walk for lung tumor segmentation from low-contrast ct images," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pp. 6071–6074, IEEE, 2013.

[9] R. Vivanti, L. Joskowicz, O. A. Karaaslan, and J. Sosna, "Automatic lung tumor segmentation with leaks removal in follow-up ct studies," *International journal of computer assisted radiology and surgery*, vol. 10, no. 9, pp. 1505–1514, 2015.

[10] M. F. McNitt-Gray, N. Wyckoff, J. W. Sayre, J. G. Goldin, and D. R. Aberle, "The effects of co-occurrence matrix based texture parameters on the classification of solitary pulmonary nodules imaged on computed tomography," *Computerized Medical Imaging and Graphics*, vol. 23, no. 6, pp. 339–348, 1999.

[11] H. Wang, X.-H. Guo, Z.-W. Jia, H.-K. Li, Z.-G. Liang, K.-C. Li, and Q. He, "Multilevel binomial logistic prediction model for malignant pulmonary nodules based on texture features of ct image," *European journal of radiology*, vol. 74, no. 1, pp. 124–129, 2010.

[12] H. Chen, J. Zhang, Y. Xu, B. Chen, and K. Zhang, "Performance comparison of artificial neural network and logistic regression model for differentiating lung nodules on ct scans," *Expert Systems with Applications*, vol. 39, no. 13, pp. 11503–11509, 2012.

[13] Q. Wang, W. Kang, C. Wu, and B. Wang, "Computer-aided detection of lung nodules by svm based on 3d matrix patterns," *Clinical imaging*, vol. 37, no. 1, pp. 62–69, 2013.

[14] J. Kuruvilla and K. Gunavathi, "Lung cancer classification using neural networks for ct images," *Computer methods and programs in biomedicine*, vol. 113, no. 1, pp. 202–209, 2014.

[15] T. Sun, J. Wang, X. Li, P. Lv, F. Liu, Y. Luo, Q. Gao, H. Zhu, and X. Guo, "Comparative evaluation of support vector machines for computer aided diagnosis of lung cancer in ct based on a multi-dimensional data set," *Computer methods and programs in biomedicine*, vol. 111, no. 2, pp. 519–524, 2013.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[17] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[18] W. B. Frakes and R. Baeza-Yates, *Information retrieval: Data structures & algorithms*, vol. 331. prentice Hall Englewood Cliffs, New Jersey, 1992.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[22] R. Girshick, "Fast r-cnn," *arXiv preprint arXiv:1504.08083*, 2015.

[23] P. Yuan, Y. Zhong, and Y. Yuan, "Faster r-cnn with region proposal refinement,"

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, *et al.*, "Going deeper with convolutions," Cvpr, 2015.

[25] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[27] R. A. Brooks, "A quantitative theory of the hounsfield unit and its application to dual energy scanning.," *Journal of computer assisted tomography*, vol. 1, no. 4, pp. 487–493, 1977.

[28] J. Mukherjee, "Local rank transform: Properties and applications," *Pattern Recognition Letters*, vol. 32, no. 7, pp. 1001–1008, 2011.

[29] M. B. Dillencourt, H. Samet, and M. Tamminen, "A general approach to connected-component labeling for arbitrary image representations," *Journal of the ACM (JACM)*, vol. 39, no. 2, pp. 253–280, 1992.

[30] S. Ramaswami, "Convex hulls: Complexity and applications (a survey)," *Technical Reports (CIS)*, p. 264, 1993.

[31] K. Briechle and U. D. Hanebeck, "Template matching using fast normalized cross correlation," in *Optical Pattern Recognition XII*, vol. 4387, pp. 95–103, International Society for Optics and Photonics, 2001.

[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.

[33] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.