Norwegian University
of Life Sciences

Master's Thesis 2022 30 ECTS
Faculty of Science and Technology

Uncertainty quantification in automated tumor segmentation using deep learning

Syed Ahmar Abbas
Master of Science in Data Science

This page is intentionally left blank.

Acknowledgements

Hereby, I would like to thank everyone for the support I received on the way to completing this thesis.

First of all, I am very grateful to Prof. Cecilia Marie Futsæther for supervising my Master's thesis and giving me great guidance by steering, stimulating, and supporting my work.

I would also like to thank Prof. Oliver Tomic for his co-supervision of this thesis, giving me insight and guidance into all data-related issues that I faced. Also, I would like to mention Ms. Bao Ngoc Huynh here, who inspired me for this project and assisted me in setting up the environment and experiments necessary for this study, and extended her support whenever asked for.

Moreover, I am very grateful to Dr. Einar Dale and Prof. Eirik Malinen for making the dataset accessible, which made this project possible, and also for their insights from the clinician's standpoint.

My special thanks go to my friends, Sanjayan Rengarajan, Alena Hensel, and others, who stood by my side during this not always easy time and supported me in every way, mentally in the form of stimulating and uplifting conversations or actively proofreading my work.

Finally, I also want to thank my family, because I know that I can always rely on them. First and foremost, my deceased parents, who sparked my interest in this field of study.

Syed Ahmar Abbas
Oslo, August 2nd 2022

Abstract

Introduction

Head and neck cancer is one of the leading causes of cancer-related deaths globally and arguably has a long-standing history of impacting human life both medically and economically. Common treatment options which are considered most effective require early and precise delineation of tumors. But this is not an easy task as it requires hours of discussion and iterations for every patient and clinical expertise, and is also prone to human error.

Due to advancements in medical imaging and deep learning, particularly with convolutional neural networks (CNN), automatic segmentation of tumors has become a hot topic for researchers, and results from different studies have shown promising outcomes. However, these auto delineation algorithms are still far from perfect, and given the nature of their use, their efficacy in the clinical environment is a hot debate. One of the reasons for the low acceptance of these CNN-based models is their indecipherable black-box nature and inability to quantify and visualize confidence in their delineations.

In this thesis, we have proposed monte carlo dropouts based technique to visualize uncertainty in the predictions of convolutional neural networks using V-net architecture to increase the interpretability of the model. This is done through visualizing uncertainties in the input feature selection of the model and also its predictions. Moreover, we have tabled a novel approach to quantify confidence in predictions with a single comparable value as a percentage for easy interpretability of the auto-segmentation to the clinicians.

Methodology

Monte carlo methods are used to obtain the probabilistic distribution of the outcomes for a numerical problem by repeated sampling. In this thesis, we are using the same approach to find uncertainties in the prediction and input feature selection of CNN-based automatic tumor segmentation model. The dataset being used for this study consists of 197 patients diagnosed with head and neck tumors. For repeated sampling, we have used dropout layers with two different rates in the model to randomly disable neurons while making predictions, this setup ensures to have slightly different predictions for the same image in each iteration. The variance in these predictions of each voxel is then visualized as an uncertainty map of the prediction, higher variance defines an uncertain region.

For input feature importance, we have used guided backpropagation which highlights only those voxels which had a positive gradient in the backpropagation pass. The approach to finding uncertainty in input feature importance is the same as for prediction, that is to find and visualize pixel-wise variance in feature selection over all samples drawn from the same patient scan.

Moreover, to quantify uncertainty in prediction, we have used the approach to estimate segmentation dice score from the overlap metric of uncertainty map and segmentation from the model. This was done by training a separate regression model, with overlap dice score from uncertainty map and prediction as the independent variable and segmentation dice as the target.

Results

Results from the input feature selection of the model concur with the previous studies on the subject done by NMBU healthcare data science group. For visualizing uncertainties, the uncertainty maps in the prediction and feature selection were found to be highlighting regions with potential false predictions with acceptable precision. Hence increasing the interpretability of the CNN model used to make predictions and assisting clinicians to focus on uncertain regions in the auto-delineated scan.

For quantification of uncertainty in prediction, our approach of using a separate regression model for segmentation dice estimation achieved an acceptable performance measured in coefficient of determination R^2.

Conclusion

The outcomes of this thesis exhibit the efficacy of using monte carlo approach to obtain uncertainty in predictions made by CNN model, hence increasing interpretability and potentially acceptability of deep learning models for automated segmentation of tumors in clinical settings.

Contents

1 Introduction **1**
1.1 Cancer: A problem for society 1
1.1.1 Auto-delineation of tumors using deep learning 2
1.1.2 Problem Statement 2
1.2 Structure of thesis 3

2 Theoretical Background **5**
2.1 Medical Imaging 5
2.1.1 Tumor Identification 6
2.2 Deep learning 7
2.2.1 Convolutional Neural Network 8
2.2.2 V-net architecture 9
2.3 Prediction Uncertainties 10
2.3.1 Methods to gauge uncertainty 11
2.4 Interpretability in convolution neural networks 12
2.4.1 Guided Back-propagation 12

3 Methodology and Experimental setup **15**
3.1 The dataset 15
3.2 Model variants 16
3.2.1 Model architecture without dropout layers 17
3.2.2 Model architecture with dropout layers 18
3.3 The training and testing procedure 19
3.3.1 Test model modifications to capture uncertainty 20
3.4 Uncertainty in prediction 23
3.4.1 Uncertainty for false predictions 23
3.4.2 Uncertainty to predict segmentation quality 24
3.5 Uncertainty in feature importance 25

4 Results **27**
4.1 Effect of dropout layers on model performance 27
4.2 Uncertainty in feature importance 30
4.3 Uncertainty in prediction 36
4.3.1 Qualitative analysis 36
4.3.2 Quantitative analysis 40

5 Discussion **45**
5.1 Importance of dropout layers 46
5.2 Qualitative and quantitative analysis of prediction uncertainty . . . 46
5.3 Visualization of uncertainties 47
5.4 Different approaches for uncertainty and future work 48

6 Conclusion **51**
6.1 Contributions of our work 51

List of Figures

2.1 Deep Learning Architecture 8
2.2 Example V-net Architecture 10

3.1 Vnet architecture without dropout layers 18
3.2 Vnet architecture with dropout layers 19
3.3 Flow diagram of end-to-end training and testing process 22

4.1 Comparison of V-net model variants using validation data 28
4.2 Patient wise performance of model variants from test dataset 29
4.3 Average test performance of model variants measured in dice 30
4.4 Uncertainty in feature importance for a patient with high overlap between error region and uncertainty map ($dice > 0.75$) 33
4.5 Uncertainty in feature importance for a patient with low overlap between error region and uncertainty map ($dice < 0.2$) 34
4.6 Uncertainty in feature importance for patient with intermediate overlap between error region and uncertainty map ($dice = 0.5 - 0.6$) 35
4.7 Visualization of Uncertainty in prediction for a patient with high overlap between error region and uncertainty map ($dice > 0.75$) . . 38
4.8 Visualization of Uncertainty in prediction for a patient with low overlap between error region and uncertainty map ($dice < 0.2$) . . . 39
4.9 Visualization of Uncertainty in prediction for patient with intermediate overlap between error region and uncertainty map ($dice = 0.5 - 0.6$) 40
4.10 Dice prediction using average uncertainty values 41
4.11 Predicted dice against segmentation dice using model.2 (with dropout rate 0.5) 42
4.12 Predicted dice against segmentation dice using model.3 (with dropout rate 0.1) 43

5.1 Recommended visualization for clinician 48

5.2 Comparison of monte carlo and ensemble method as Bayesian approximation 49

List of Tables

3.1 Count of patients and folds belonging to each of the datasets 16
3.2 Model variants for training with varying dropout and learning rates 20
3.3 Regression models to predict segmentation dice score 24

4.1 Results from model variants for the training and validation process 29
4.2 Patient scan slices with high ($dice > 0.75$) overlap between error region (ER) and uncertainty maps. 31
4.3 Patient scan slices with low ($dice < 0.2$) overlap between error region (ER) and uncertainty maps. 31
4.4 Patient scan slices with intermediate ($dice > 0.5$) overlap between error region (ER) and uncertainty maps. 31
4.5 Model performance measured in the overlap between error region and uncertainty map 36
4.6 Regression models to predict segmentation dice score from the overlap of segmentation and uncertainty map 42

Abbreviations

Abbreviation	Meaning
API	Application Programming Interface
BN	Batch Normalization
Conv	Convolution
CNN	Convolutional Neural Network
CT	(X-Ray) Computerized Tomography
HDF(5)	Hierarchical Data Format (5)
IO	Input/Output
JSON	JavaScript Object Notation (a standard data serialization format)
MC	Monte Carlo
PET	Positron Emission Tomography
PPV	Positive Predictive Value (i.e. precision)
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
SGD	Stochastic Gradient Descent
SGDR	Stochastic Gradient Descent with Warm Restarts
STD	Standard Deviation

Chapter 1

Introduction

1.1 Cancer: A problem for society

Worldwide, cancer is one of the main causes of death, responsible for nearly 10 million deaths in 2020 [1]. A rising age expectancy in the global population is identified as a leading cause of an increase in cancer cases, which foresees an estimated amount of over 16 million deaths caused by cancer in 2040 [2]. Head and neck cancers(HNC) are one of the most common forms of cancer. Under this umbrella term HNC, tumors in the oral cavity, pharynx, lip, larynx, and paranasal sinuses; occult primary cancer, salivary gland cancer, and mucosal melanoma are considered [3].

There are several treatment methods for cancerous diseases, also referred to as malignant tumors or neoplasms, which include Surgery, Radiation therapy, Chemotherapy, and Targeted therapy, all of which are dependent on the delineation or contouring of tumors and are only successful if done in an early stage [4]. Tumor segmentation involves correctly identifying the location of a tumor within a medical image.

Although this has historically been done manually, manual tumor delineation is a time-consuming process and is usually performed by experienced radiologists. Given the time complexity of cancer treatment and the accuracy required for lesion delineation, computer-assisted automated segmentation is considered a viable solution to reduce the time required for tumor delineation and increase accuracy [5].

1.1.1 Auto-delineation of tumors using deep learning

Since segmentation of tumors from medical images is an image classification problem, many studies based on using deep learning have shown promising results [6][7][8] since the advent of convolutional neural networks [9]. Previously, image analysis-based heuristics were used to automatically segment tumors from medical images which included algorithms such as thresholding, edge detection, clustering, watershed, etc [6] but the use of convolutional neural networks increased the accuracy of segmentation by many folds.

Convolutional Neural Networks (CNN) is a particular type of neural network which are specifically designed to cater to image segmentation problems [8] as they utilize more spatial information from images as compared to histogram-based approaches. However, like any other AI algorithm, they are also not perfect and make false predictions.

Given the sensitive nature of tumor delineation, which requires high accuracy for precise delivery of treatment medicines, the adoption of convolutional neural networks in clinical environments has always been in question, primarily due to the inability of CNNs to provide confidence and interpretability in its predictions.

1.1.2 Problem Statement

The use of machine learning in medical imaging applications has come a long way [10][11] but its full adaptation with confidence in the clinical environment is still far from its final goal. Particularly after the development of deep learning algorithms to automatically segment tumor volumes, which not only saves time for the delineation of tumors but also provides high accuracy by removing the human factor, the debate for its acceptance and adoption in clinical circles is still ongoing.

Although these algorithms have shown promising performances for a large set of patients, they are still prone to failure for some and there could be multiple reasons contributing to these failures including but not limited to, the dataset used for training, location, and the type of tumor, the quality of the hardware used for input scans and modalities involved.

But the reluctance in their adoption in clinical environments is not only limited to their propensity to fail, but also because of the indecipherable black-box nature

of these deep learning models. Some of the challenges faced by medical experts to adopt deep learning-based segmentations which, we have tried to address in this thesis are summarized here as:

- Interpretability in convolutional neural networks. As, CNNs lack this property inherently, which is the ability to produce and visualize the features in the input scan which were considered important by the model to make a particular prediction. Also, those features where the model was not certain of their importance. This property is crucial because it might help radiologists to compare feature importance given by the model based on their expertise in the medical field.
- Uncertainty in predictions, the ability to quantify the probable errors in its predictions and also visualize those regions where the deep-learning model was unsure of its predictions. This property is important to assist radiologists to give importance and focus more on the regions where the model is unsure of its predictions.

1.2 Structure of thesis

After the introduction, in this thesis, we have discussed the background related to our problem statement in detail. This includes a brief discussion and literature review of medical imaging and how deep learning can assist in delineating tumors from medical images automatically. We have also discussed briefly deep learning algorithms that are commonly used for tumor segmentation, convolutional neural networks in particular, and also discussed components involved in a typical convolutional neural network model.

Since the model used in this study is based on V-net architecture, for 3D medical images. We have included a short literature review of this architecture as well. Moreover, in the theory, we have discussed the definition of uncertainty in predictions of the neural network model and how the monte carlo method can be used to achieve that. Post theoretical background, we have discussed our methodology which includes experimental setup as well. In the end, we have shared results from the study followed by discussion and recommendations.

Chapter 2

Theoretical Background

Working on this thesis involves multidisciplinary knowledge. So the following pages contain the background knowledge related to the thesis. First, we look at the process of obtaining details about the internal structures of the human body through non-invasive or mildly invasive medical techniques. Then we look into what makes a tumor different in these scenarios. After getting a clear picture of the medical imaging process, the automatic detection utilizing deep learning with its components, usage, and limitations are further discussed.

2.1 Medical Imaging

To properly diagnose certain diseases, we need to understand what is the current state of the human body. This state, when compared to a healthy population, we can conclude the differences as the issues regarding the disease. Although this is an extremely simplified way of explaining the process of diagnosis, the fundamental process holds truth. To get details about the state of the human being one can observe the human visually or also can orally verify the issues experienced by the human. But, this process limits the diagnosis to diseases that exhibit symptoms and have some external visual cues. To extend diagnostic power to a wide variety of diseases and to perform early diagnostics, it would be best if the internal state is known to the doctors. This state can be measured through various techniques but can be classified broadly into invasive and non-invasive techniques.

Medical Imaging is a type of non-invasive technique, which provides more insight

into the internal state and structure of the human body. Although the majority of the medical imaging is performed non-invasively, sometimes there are mildly invasive techniques that are performed to obtain suitable results. Some such example is the injection of specialized dyes which can improve contrast while imaging. As the medical imaging that needs to be performed is done to find the internal state, the imaging process can not be performed in the visible spectrum of Electromagnetic Waves (EM). This means the imaging is done in the other spectral part of the EM wave. For example, the imaging techniques for Computed Tomography (CT) scans utilize the spectral range corresponding to EM Waves, and Magnetic Resonance Imaging (MRI) scanning system is based on the radio wave spectrum. Positron Emission Tomography or PET scanning technology utilizes the gamma wave spectrum of the EM waves. With modern advancements in medical technologies, the results of performing a scan or imaging have higher resolution and with the support of an image processing algorithm give a 3d output with precise measurements.

Doctors are trained to read the results of the scan, with some doctors even specializing to interpret particular results of the scan and correlate them with the list of known issues to arrive at a conclusion for a diagnosis. Apart from diagnosis, the imaging techniques also provide a visual aid for surgeons to prepare themselves for safe surgery. Since in our thesis we mostly worked on tumor segmentation, we look in the following chapter how tumors are identified by medical professionals from the results obtained from the scanning process.

2.1.1 Tumor Identification

Tumors are nothing but cells that often fail to perform their regular function and multiply at a rate much more rapid than regular cells. This makes tumors compete for the nutrition and oxygen required by the healthy cells performing the actual function. Since tumors multiply rapidly they often make the healthy cells starve and suffocate, thereby killing them in the process [12]. If the death of healthy cells eventually happens on a larger scale as the tumor grows, thereby making very few cells left to perform the bodily functions necessary for survival. This is the cause of fatality or severe sickness among cancer patients as the tumor progresses.

The identification of tumors from medical imaging can be even quite arbitrary in the final stages of tumor growth for certain types of cancer. Since a tumor would appear different than what would be considered a normal internal structure a large tumor can have distinctive features which can highlight it from the rest of

the cells. As often we do not want to wait till the final stages of the disease to start with treatment for early diagnosis of tumors, we can differentiate a tumor cell from the healthy cells by looking at the consumption of nutrients or the blood flow patterns to that particular area. Tumors also have different absorption, so using a specialized dye injected into the bloodstream might eventually accumulate in a larger concentration within the cells of the tumors. This dye can then be scanned in the human body looking for its concentration gradient across the body.

2.2 Deep learning

Realizing the merits of intelligence in form of computation has been a focus for researchers, even from the beginning of the computational era. With the shrinking of processor size and the number of cores that can fit in a single die, the power of parallel computing has provided a huge boost for this field. Apart from the computational power, with the advent of digitalization, the availability of data also exploded. A major part of artificial intelligence is to make the machine learn itself by training it with input data and the expected output data. Therefore, the increment in the amount of data has resulted in more reliable training. Deep learning is one of the emerging fields of machine learning models. The concept of deep learning draws inspiration from the biological process of learning. As how in the human brain the learning processes are significantly influenced by the neurons and their connection strength, in deep learning the same concept has a simplified mathematical model of the biological process.

The architecture for a deep learning model as shown in Figure 2.1, consists of a repeated number of layers connected in a successive manner. The starting layer or the input layer is the layer containing nodes that receive the input data. This input layer has the same dimensions as the width of the input data. After passing through this layer the data is multiplied by the weight value of the connection and shifted by a bias. Then the resultant values are applied to a function called the activation function and those are the values that get stored in the subsequent node in the hidden layer. This mathematical process continues cascading till the final output layer. The final value in the output layer is then taken to be the output data. The model's parameters such as individual connection weights and biases constantly get updated in the training phase of the model, with the update proportional to the ratio of expected value to that of obtained value. The mathematical process through which the network's parameters get updated during training is referred to as back-propagation [7].

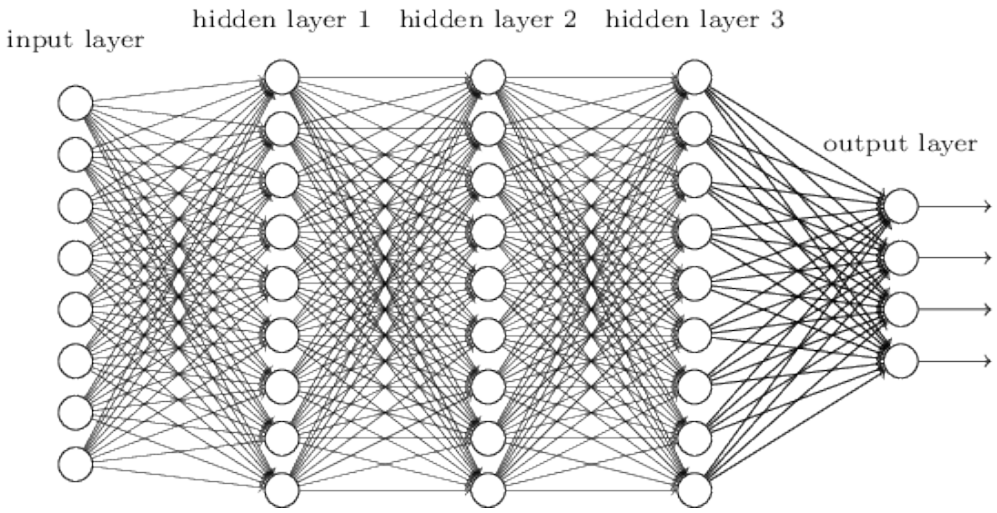


Figure 2.1: An architectural diagram of the deep learning network. The circles represent the nodes and the arrows represent the connection between different nodes. Here only 3 hidden layers are represented, but usually, there are many layers of hidden layers before the output layers. Note: Adapted from [13], permitted usage under Creative Commons license.

2.2.1 Convolutional Neural Network

Convolutional Neural Networks are a type of Artificial Neural Network (ANN) that specializes in image classification. The architecture of CNN was designed based on inspiration from the biological neural system[14]. A typical CNN has three base layers [15],

- Convolutional layer
- Pooling layer
- Fully-connected layer

The convolutional layer can be considered the backbone of CNNs. The purpose of the convolutional layer is to find important features from the input image. This is done by sequentially moving a kernel or filter over the receptive field of the input image to find important features by taking the dot product of the input pixel and kernel pixels. The final output (commonly known as a feature map) is actually an array of these dot products.

The next is the pooling layer, also called the downsampling layer. The sole purpose of this layer is to reduce the shape of its input image. This is done in a similar fashion as in the convolutional layer, by running kernel over input, however, here only aggregation is done. This aggregation can either be 'max pooling' which is selecting the maximum value of the input pixels or it can be 'average pooling', which is by taking the average of input pixel while sliding kernel over the input image. This is done to reduce the computational complexity of the model.

In fully-connected layers, unlike partially connected layers, each output node is directly connected to the node in the input layer. This layer is used to perform final classification and typically uses a softmax activation function to produce class probability.

In addition to these layers, there are other layers as well that can be used based on requirements. One such notable layer is the dropout layer, which disables nodes from the previous or input layer to regularize the model. This is usually done in the training phase and these layers are kept disabled while making predictions.

There are different architectures that have been developed on top of convolutional neural networks to solve a different set of problems which includes U-net, ResNet, V-net, nn-Unet, etc. In this thesis, we have used V-net architecture to segment tumors in PET/CT scans and to quantify uncertainties in predictions.

2.2.2 V-net architecture

V-net architecture is very similar to the infamous U-net CNN architecture [16], however, the only notable difference is that V-net takes more spacial information from the input scan as it performs 3D convolutional operations on 3D input images.

A typical V-net architecture consists of encoder and decoder sections, each of these sections can have multiple blocks. In the encoder, each block contains 3D convolutional layers to get feature maps followed by a max pooling layer to reduce the size of the input image. While the decoder has all inverse operations as of the encoder, it contains convolutional layers followed by a 3D up-convolutional layer, to restore the segmented image to the original image shape. In addition, at each block, feature maps from the corresponding encoder layer are appended to reconstruct the final segmentation.

Figure 2.2 below shows an example of V-net architecture, the image has been used under Creative Commons license.

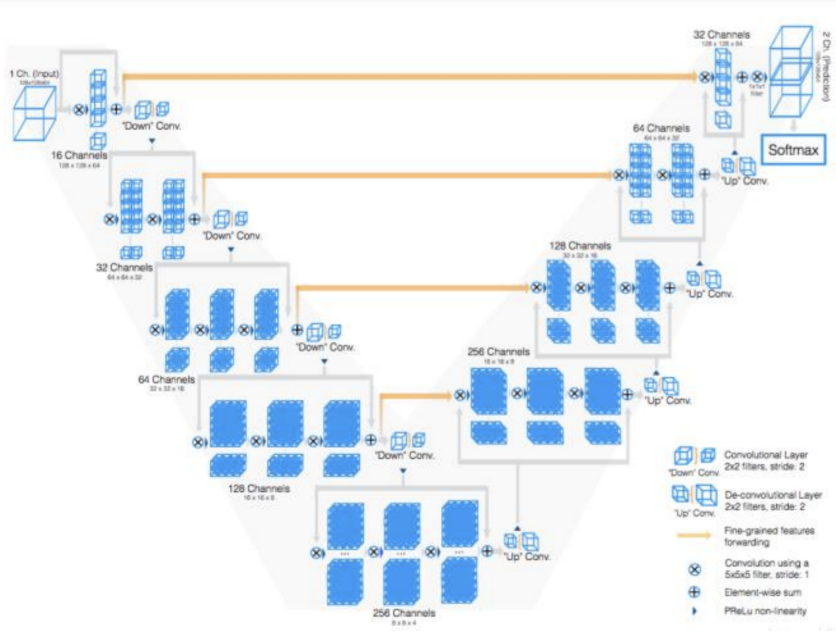


Figure 2.2: An example V-net architecture, shows the down-sampling path on the left and up-sampling path on right. Adapted from [17], permitted usage under Creative Commons license.

2.3 Prediction Uncertainties

A CNN model, although it is among the forefront runners for successful predictions and classification of tumors, they come along with quite some drawbacks. The major drawback is that the results of predictions do not cover the detail regarding the confidence level for that prediction. This is also one of the problem statements that this thesis work seeks to address. The confidence level of prediction in simpler terms can be explained as when the model predicts a certain region to have a tumor, it does not always have complete certainty on its prediction. As the usage of the automated diagnostic systems is only to enhance and support the diagnostics made by a medical professional, without knowing the reliability of the prediction, the doctors might have hard distinguishing which predictions are reliable. If the doctors then have to virtually check every prediction with the same amount of effort, it defeats the purpose of the system.

The uncertainty if quantified and presented could indicate the reliability of the prediction. As no model is 100 percent accurate, if they come along with a measure

of confidence in their predictions, it could aid in improving the diagnostic efficiency.

2.3.1 Methods to gauge uncertainty

Quantifying uncertainty in prediction is challenging, requiring a strong statistical model. Bayesian inference[18] is one such method, which has been employed for this task [19]–[21]. The equation describing this method of inference is given in Equation (2.1).

$$P(y_*|x_*, \mathfrak{D}) = \int P(y_*|x_*, W) \cdot P(W|x_*, \mathfrak{D}) dW \tag{2.1}$$

where,

W = The wights of the model
$\mathfrak{D}$ = The dataset containing the images and the tumor segments
x_* = A new data sample of the medical images
y_* = A new data sample detailing the tumor segments in x_*

It is easier to calculate the first factor in the integral, as it could be done with a single pass, but in a bayesian neural network calculating the second part of the integral is quite difficult. This is because the equation requires calculating the distribution of weights which can not be done analytically. This makes the cost of computation infeasible in practical terms. So to have a workaround for the same, other approximations for the calculations are employed. A dropout method [22] is one of the methods that have been used. In a dropout method, some of the connections in the neural networks are as the name suggests dropped out. Since, if the dropped-out connections are the same during data collection, it would not result in variation in data. So it should be randomized, this is the process called Monte Carlo dropout [23] and is the method that we have utilized in this thesis. When this method is used then the value can be approximated as given in Equation (2.2).

$$P(y_*|x_*, \mathfrak{D}) \approx \frac{1}{T} \Sigma_{t=1}^{T} F(f_{W_i}(x_*)) \tag{2.2}$$

where,

T = Total number of samples
F = Softmax function to calculate the probability
f_{W_i} = The model f as a function along with its weights parameters

The variance among different tumor segmentation predicted by the models gives us the value of the uncertainty for the model.

2.4 Interpretability in convolution neural networks

For doctors to understand a prediction, it should be in such a form that its predictions can be backed by some scientific reasons. Since CNNs are designed only to provide predictions, they make it harder for medical professionals to trust and interpret the predictions. if the predictions even have to a degree what regions of the image had made it arrive at this conclusion, then the predictions can be trusted to a greater extent, if those specified regions seem reasonable to the professional [24].

One of the methods which have been observed to have a good effect on the same is through guided back-propagation which is explained in the following section

2.4.1 Guided Back-propagation

Guided back-propagation [25], [26] involves analyzing the gradient present in the deep learning network with respect to the input image. A high positive gradient implies the particular pixel has high importance to that of prediction. Conversely, a negative gradient implies very low importance. Including both positive and negative gradients will not properly highlight the important region as they would be noisy. In this process, the negative gradients are equated to zero and only the positive gradients are kept as is. This is done in a backward pass of the neural network.

Using a similar Monte Carlo method as discussed above, the confidence value for each pixel impacting the decision can be calculated as given in the Equation (2.3)

$$q(\sigma^0|x_*) \approx \frac{1}{T}\Sigma_{t=1}^{T}\nabla_\theta f^{gb}(x_*; W_t^*) \tag{2.3}$$

where,

T = the total number of samples
σ^0 = The specific pixel or feature of the image
q = The significance of a pixel/feature for the segmentation
∇_θ = The gradient of the layer w.r.t the parameters

Chapter 3

Methodology and Experimental setup

Methods to quantify uncertainty in the segmentation made by deep learning models, that were briefly discussed in Chapter 2 were implemented in this thesis using a dataset that contains 3D scans of head and neck cancer masses.

This chapter highlights significant details of the dataset that was used for the study. Since this thesis is a continuation and extension of the work of the healthcare research group at NMBU, more granular details about the dataset can be found in past research papers and dissertations [16][7].

Furthermore, this chapter explains the experiments that were performed and describes the implementation of theory from Chapter 2 for quantification and visualization of uncertainties in auto-delineation of tumors using convolutional neural network models for interpretation.

The code to perform experiments and analysis is available at GitHub repository (https://github.com/ahmarabbas14/MS_uncertainity) and can be used to replicate results.

3.1 The dataset

The dataset provided for this thesis was contained in a single HDF5 file. It contains 3D scans from the head and neck regions of 197 cancer patients who went through

treatment at Oslo University Hospital, the Radium hospital. The scans include both CT and PET modalities [7].

The dataset also contains manual delineations from experienced radiologists at the Oslo University Hospital, corresponding to each patient's PET/CT scan. These delineation masks included gross tumor volume (GTV) and also the affected lymph nodes for the patients with advanced stage and were used as ground truths for deep learning models in this study. In cases where multiple delineations were available, a union of these delineations was used as ground truth. Moreover, a voxel in the ground truth image for a patient that represents a healthy (or non-cancerous) tissue has a value 0 while a cancerous voxel is marked as 1, which makes this a binary segmentation problem.

The HDF5 file contains fourteen groups in the file root, each group has dataset "input" (PET/CT scans) and "target" (manual delineations). Each of these groups represents a fold that belongs to repurposed dataset after splitting for training, validation, and testing purpose as shown in Table 3.1

Table 3.1: Count of patients and folds belonging to each of the datasets

Dataset	Folds	No. patients
train	0-9	142
val	10	15
test	11-13	40

More details about the dataset can be found in the previous thesis from the same research group in Moe's work [7] and Huynh's thesis [16].

For this thesis, methods for visualization and quantification of uncertainty in segmentation were applied on the folds belonging to the test dataset as explained in Table 3.1. Also, it is important to note here that this data was stratified based on tumor stage to avoid any bias.

3.2 Model variants

As explained in the previous section, the dataset we have available has three dimensions, to utilize the full potential of convolutional neural networks for 3D

data [27], we trained this dataset on different variants of volumetric convNet (V-net) architecture [28]. V-net architecture has become a popular choice for medical image segmentation problem recently, as it requires pixel-wise classification of the 3D input image and 3D images provides more spatial signal as compared to 2D [29][30].

For this thesis, first, we trained and compared four different models to measure the effect of using dropout layers during the training process. Details of these models are highlighted in Table 3.2.

During the testing phase, model.2 and model.3 as mentioned in Table 3.2 with dropout rates 0.5 and 0.1 were used to make different predictions in multiple iterations, details of the testing process and method to visualize and quantify uncertainty in this thesis work are mentioned in Section 3.3 and Section 3.4 respectively.

Performance metrics

Dice and Jaccard Index are often considered reliable metrics to evaluate the performance of medical image segmentation problems [31]. In this thesis, the Dice score was used to analyze and compare the performance of different models and outcomes. Dice score (also known as Sørensen–Dice index) is calculated in a similar way as Intersection-over-Union (IoU) but by giving twice the weight-age to correctly identified pixels or voxels. The equation for calculation of dice score is given below Equation (3.1)

$$Dice = \frac{2TP}{2TP + FN + FP} \tag{3.1}$$

Where TP stands for True Positive, which is the count of voxels that were tumorous and were predicted as such. FN represents False Negatives, the count of voxels that were incorrectly classified as non-cancerous and FP stands for False Positive, which is the total voxels that were incorrectly predicted as cancerous.

3.2.1 Model architecture without dropout layers

The base V-net model which was used in this thesis consisted of the encoder (or down-sampling) and the decoder (up-sampling) sections, each of these sections has four blocks. For the encoder section, In each block, there are two 3D-convolutional

layers with a kernel size of 3, followed by a 3D max-pooling layer. The filters in each convolutional layer double after every block, starting from 32 and going all the way up to 256. This architecture makes sure that the feature map is doubled after every block and the input shape is reduced to half.

For the middle block (bridge) between encoder and decoder sections, two 3D-convolutional layers with 512 filters and 3 kernel size are followed by a 3D up-convolution layer.

In the decoder section, just like the encoder, two 3D-convolutional layers are used at every block followed by an up-convolutional layer. In addition, the feature maps from the corresponding encoder section are appended to the input of the layer to reconstruct the segmented image. In the end, activation from a sigmoid function is used to give the probability of a voxel belonging to either of the classes. Figure 3.1 gives a visualization of the architecture explained here.

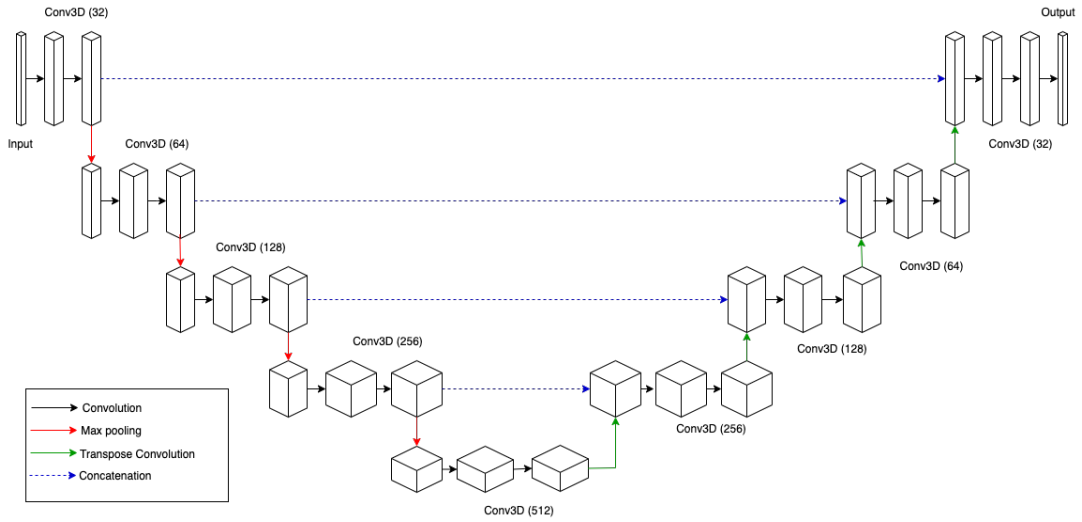


Figure 3.1: Visual representation of the variant of Vnet architecture which was used in this thesis. Convolution layers are marked with labels and the number of filters in each layer is marked in the parenthesis. Layers between input and Conv3D(512) mark the Encoder or the down-sampling path on the left and the rest is the up-sampling path

3.2.2 Model architecture with dropout layers

Core architecture for the model with dropout layers is the same as the one explained in Section 3.2.1 to a larger extent. However, two dropout layers were

added in the middle or bridge block after each 3D-convolution layer. This architecture was trained and tested with two different dropout rates, 0.5 and 0.1 respectively as mentioned in Table 3.2.

Figure 3.2 below depicts the placement of these dropout layers in the architecture. The placement of the dropout layers here was inspired by Wickstrøm's work [26]. However, during the experimental setup, another architecture was tried with a dropout layer after every convolutional layer in encoder block, which increased the complexity of the model exponentially and we did not have enough resources to train such model, hence the idea was deemed not feasible to implement.

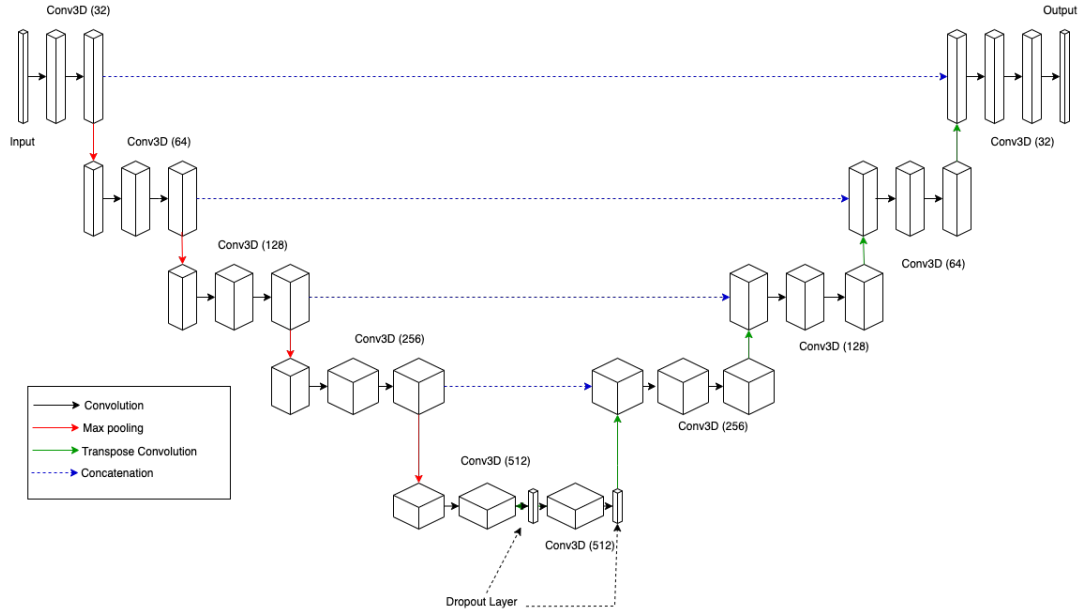


Figure 3.2: Another variant of Vnet architecture as discussed in Figure 3.1, However, this variant marks the dropout layers with their placement. This architecture was used with different dropout rates as mentioned in Section 3.2.2

3.3 The training and testing procedure

The 'train' and the 'validation' datasets as explained in Table 3.1 were trained and validated on the four different models as shown in Table 3.2 to analyze the effect of adding dropout layers in the model for strong regularization. The mean of dice scores from all patient scans in the test data was used as the comparative metric in this thesis.

Table 3.2: Model variants for training with varying dropout and learning rates

Model	dropout rate	learning rate
model.1	0	0.0001
model.2	0.5	0.0001
model.3	0.1	0.0001
model.4	0.1	0.001

During the training process, trained weights were saved in an HDF5 file after every 5th epoch and the epoch with the best performance (minimum loss) was marked to be used at the time of testing.

To estimate the predictive distribution and input feature importance using Monte Carlo dropout method, the best epoch weights from the pre-trained model 'model.1' as mentioned in Table 3.2 were used. These weights were loaded at the test time to the models 'model.2' and 'model.3' described in Table 3.2 and 20 iterations were made on each of the models with the 'test' dataset scans, that gave us 20 different predictions for each of the scan in the dataset and 20 different input feature importance corresponding to those predictions.

The output predictions and guided back-propagation based feature importance from each iteration were stored in a separate HDF5 file and were further processed as explained in Section 3.4 and Section 3.5

All experiments which were mentioned here were performed on Orion, High-Performance Computing (HPC) cluster provided and maintained by the IT department of NMBU, using shared Graphics Processing Units (GPU) [32]. At a time, four experiments were executed in parallel at max and they took on an average between 12 to 18 hours to complete based on the load on shared resources.

Figure 3.3 defines process flow for training and testing using Monte Carlo dropout method.

3.3.1 Test model modifications to capture uncertainty

Adding dropout layers to a model is an effective method to regularize neural networks and reduce overfitting [33]. However, as per TensorFlow's documentation,

these dropouts by default are only applied in the training phase and are kept disabled at the time of testing or making predictions. Monte Carlo based testing models as explained in Section 3.3 require dropout layers to be activated and applied at the time of making prediction as well. Hence, the dropout layers in the test model call a custom API of deoxys framework [16] which keeps dropouts active using the 'training' call argument enabled as per TensorFlow's documentation [34]

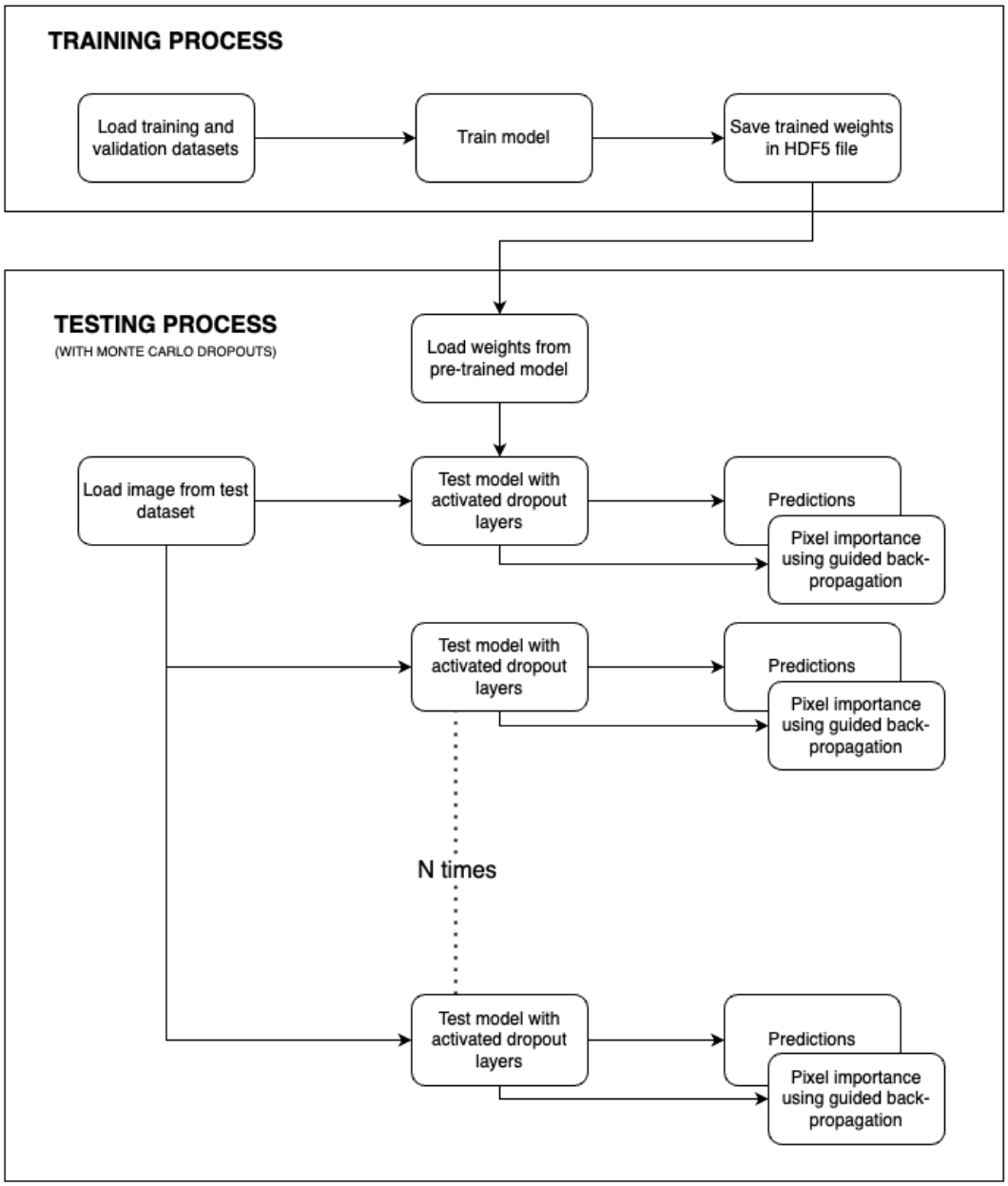


Figure 3.3: Visual representation of end-to-end training and testing process. Each box identifies a process and the arrow show flow of the process. During testing with Monte Carlo methodology, each image was given a forward pass with the model 20 times (N=20) to estimate predictive distribution

3.4 Uncertainty in prediction

As defined in Section 3.3, to obtain uncertainty in prediction, 20 different predictions were made on each of the test scans using Monte Carlo Method and these predictions were stored in an HDF5 file. They were further processed to produce an uncertainty map. The uncertainty map highlights the voxels in segmentation on which the deep learning model was not sure of its prediction.

This was done by stacking all predictions of a particular scan and taking standard deviation over each of the voxels. The voxels with higher standard deviation represent an uncertain region, however, the voxels where the model was very sure of its prediction should theoretically have very similar or close class probability values hence lower standard deviation. The standard deviation value for a particular voxel is referred to as the uncertainty value here.

These uncertainty maps were thresholded to remove noise, this was done by setting all voxels which had uncertainty values lower than the mean of all uncertainty values plus thrice the standard deviation of all uncertainty values for a particular scan to zero. The purpose of thresholding the uncertainty map is to highlight only those voxels where the deep learning model is really unsure of its prediction and remove all very small uncertainty values which actually show confidence in the model.

The thresholded uncertainty maps were further plotted using the python library matplotlib [35] to visualize uncertainty in prediction.

Moreover, the mean of all predictions for a particular scan was used as the final segmentation in this thesis work.

3.4.1 Uncertainty for false predictions

In order to quantify the effectiveness of uncertainty maps to highlight false predictions of the deep learning model, it was important to find regions where segmentation was done wrong. This included all voxels with False Positive and False Negative predictions.

This was done by voxel-wise subtraction of ground truth and deep learning segmentation. The absolute difference was stored and displayed as an error region (ER) for qualitative and quantitative analysis in this thesis. The overlap of error

region and uncertainty map was summarized using dice score Section 3.2, higher the dice value indicates better identification and representation of error region by uncertainty map.

3.4.2 Uncertainty to predict segmentation quality

Another problem based on the indecipherable black-box nature of deep learning models which was explained in earlier sections is that it does not give a measure of confidence in its prediction [36][37]. Only visualizing thresholded uncertainty map with the auto-segmentation from deep learning models does not fulfill the purpose as this in practice is not quantifiable, and its interpretation may vary from clinician to clinician.

To address this issue, based on the recommendation and work from [36] we quantified segmentation quality based on the overlap between uncertainty maps and auto-segmentation made by the deep learning model. This overlap was measured in dice score Section 3.2, the high dice score, in this case, means higher uncertainty in prediction. Dice scores obtained from the overlap of the uncertainty map and segmentation from the test dataset were further trained using four different shallow learning regression models using actual prediction dice score as the target to predict segmentation dice score in real-time.

Table 3.3: Regression models to predict segmentation dice score

Regression Model	Model Class	Hyper-parameters
GradientBoostingRegressor	Ensemble	$random\_state = 0$
RandomForestRegressor	Ensemble	NA
LinearRegression	Linear	NA
Ridge	Linear	$alpha = 0.5$

These models were trained with 60% of the data from the segmentation output in the test dataset and were tested with the whole test dataset (i.e. 40 patients)

3.5 Uncertainty in feature importance

Similar to the process of visualizing uncertainty in prediction, input feature importance produced using guided back-propagation approach was stored in an HDF5 file as mentioned in Section 3.3. This data included two modalities of scans PET and CT and were stored as channels in the output file. At the time of visualization, both these modalities were displayed on top of each other.

To visualize the uncertainty in feature selection by the model for prediction, all gradients depicting voxel importance for a particular scan were stacked and the voxel-wise standard deviation was calculated to visualize uncertainty in feature importance. The higher standard deviation at a voxel represents that the model was not sure if that particular voxel was important for classification.

The voxel-wise mean was used to depict actual feature importance which was given to each voxel by the model to produce segmentation and was visualized in a different color than uncertainty.

Chapter 4

Results

The methodology for the implementation of monte carlo dropouts based approach to find uncertainty in predictions and input feature importance as detailed in Chapter 3 was implemented using the *deoxys* framework developed by Huynh [16].

In this chapter we are going to discuss results from the experiments that were performed, we have divided this chapter into 3 main components. First, we are going to share results from the comparison of model variants that differ from each other based on the use of dropout layers and learning rates as mentioned in cite table. Followed by a qualitative analysis of results from visualization of input feature importance and uncertainty associated with it. And finally, in the end, results from visualization of uncertainty in predictions and performance of our approach to predict segmentation quality based on overlap metric between uncertainty map and deep learning based segmentation are discussed.

4.1 Effect of dropout layers on model performance

As explained in Section 3.2, four different model variants were trained to measure the effect of using dropout layers in the architecture. Out of these four variants, three were trained using dropout layers while the fourth one was 3D replication of the architecture from Huynh's work [16] and was used as the benchmark.

During the training process, the model was validated using validation dataset after every 5 epochs. Figure 4.1 shows dice score from validation dataset after every 5th epoch when validation was performed. Since early stopping with a tolerance of '30' was used during the training process, hence we can see model.1 and model.3 stopped training after 80 and 85 epochs respectively.

The Figure 4.1 also shows that the maximum dice score (0.659) during validation was achieved by model.2 at the 100th epoch. In addition, the line plot shows model.3 has the most consistent performance over epochs.

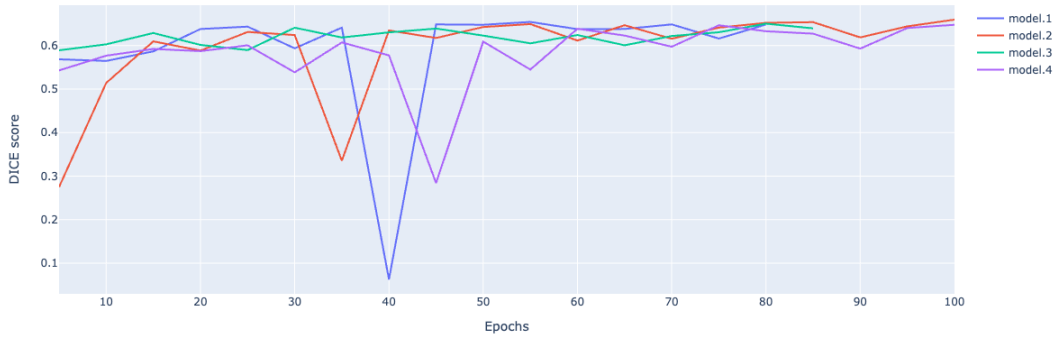


Figure 4.1: Scatter plot depicting performance of model variants as explained in Table 3.2 on validation dataset. Validation was performed after every 5 epochs. Since early stopping was used during the training and validation process, hence model.1, model.2, model.3, and model.4 stopped training after 80, 100, 85, and 100 epochs respectively.

Table 4.1 shows average dice score performance and standard deviation in dice score over epochs for all model variants. Although the average dice performance does not change significantly between variants, the results reflect that all model variants with dropout layers outperformed the benchmark model without the dropout layers. In addition, the standard deviation in dice score over epochs for model.3 is significantly low which shows that its performance is rather stable and does not change a lot over epochs. The low standard deviation over epochs for dropout-based architectures, explains and complies with the theory of using dropouts for stronger regularization and preventing models from over-fitting as explained in Chapter 2

Table 4.1: Results from model variants for the training and validation process

Model	No. of epochs	Avg. dice	Standard deviation in dice over epochs
model.1	80	0.589	0.143
model.2	100	0.593	0.103
model.3	85	0.619	0.018
model.4	100	0.585	0.078

After training and validation, all model variants were tested using the test dataset as mentioned in Table 3.1. Figure 4.2 shows the performance of each variant on the test dataset measured in dice score. One obvious observation from the line graph (Figure 4.2) below shows that all models performed worse for patient IDs 110, 82,93, and 16, however, 'model.3' appears to perform relatively better in most of the cases. The worse performing patients were manually verified and it was found that there was not enough signal in PET/CT scans for these patients. This observation goes in line with studies based on the same datasets from the NMBU healthcare data science group [7][16].

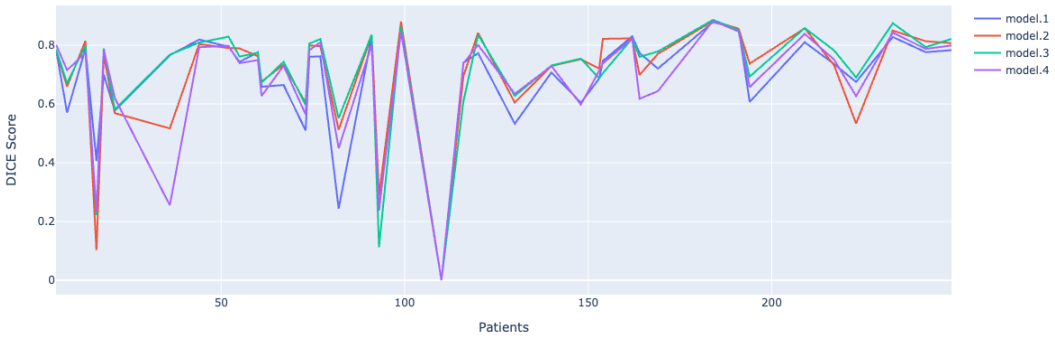


Figure 4.2: The plot shows performance of model variants (Table 3.2) on test dataset. The horizontal axis marks patient IDs in the test dataset, while the vertical axis shows their respective dice score.

Figure 4.3 below shows the average dice score performance of all model variants that were experimented with. model.3 with the dropout rate of 0.1 and learning rate of 0.0001 outperformed all other variants with an average dice score of 0.708.

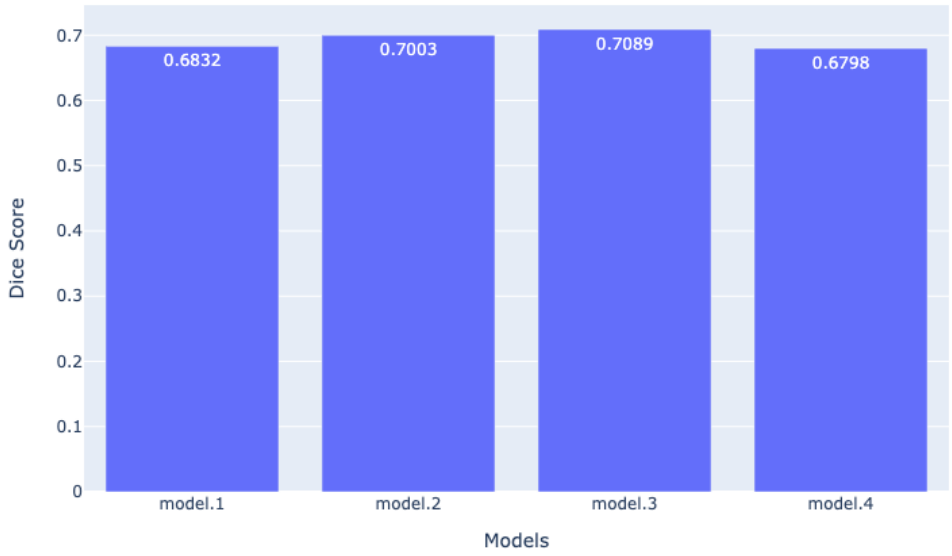


Figure 4.3: Illustrates average dice score on the test dataset of the training model variants. 'model.3' (0.1 dropout rate and 0.0001 learning rate) outperforms others with an average dice score 0f 0.7089

Although the average dice performance did not increase significantly with the use of the dropout layers, from comparative results before, it is easier to deduce that dropout layers do regularize the model and prevent over-fitting.

4.2 Uncertainty in feature importance

Feature importance for predictions in each iteration of monte carlo approach was calculated using guided backpropagation. The voxel-wise mean was used to visualize features that were considered important for prediction by the model while the variance in the gradient of input voxel was used to visualize uncertainty in feature importance. Higher the variance shows that model was unsure about the importance of that particular voxel for making predictions.

For further qualitative analysis of uncertainty in feature importance and predictions following groups of patients were selected, these groups were formed based on slices with high($dice > 0.75$), low($dice < 0.2$) and intermediate($dice > 0.5$

and < 0.6) overlap between error region (ER) and uncertainty maps as shown in Table 4.2, Table 4.3 and Table 4.4

Table 4.2: Patient scan slices with high ($dice > 0.75$) overlap between error region (ER) and uncertainty maps.

Patient_idx	*slice_idx*	dice
110	58	0.813
169	34	0.766
242	121	0.763

Table 4.3: Patient scan slices with low ($dice < 0.2$) overlap between error region (ER) and uncertainty maps.

Patient_idx	*slice_idx*	dice
164	81	0.161
191	63	0.142
120	136	0.131

Table 4.4: Patient scan slices with intermediate ($dice > 0.5$) overlap between error region (ER) and uncertainty maps.

Patient_idx	*slice_idx*	dice
73	94	0.510
116	75	0.519
16	57	0.572

Figure 4.4 Figure 4.5 and Figure 4.6 are the visualization of importance and uncertainty of importance of input features for making predictions. These visualizations were created based on patient groups selected for analysis as shown in Table 4.2 Table 4.3 and Table 4.4 respectively.

All these visualizations have four columns, from left to right, the first column contains PET/CT scan with ground truth and prediction contours, the second

column contains the uncertainty map of predictions, and the third column shows feature importance and uncertainty in feature importance in CT channel and the last column shows the same in PET channel. For feature importance, pixels highlighted in green define the region where the model was sure that those features are important for prediction while voxels marked in red highlights voxel where the model was unsure of their importance.

From Figure 4.4(a) we can see in the image labeled as PET/CT, that there is some part of gross tumor volume (GTV) that was correctly segmented while the rest of the GTV was not. However, when combining this with information from the uncertainty map and feature importance from both modalities, we can see that the false negative region has an uncertainty mark for those pixels, also feature importance in both modalities shows uncertain features in the same region.

Figure 4.4(b) is a good example of false positive prediction, we can see that the error dice for the slice is 0.813 which is a high number (on a scale of 0-1) and the uncertainty map surrounds the segmented region. When we compare the uncertainty region in prediction to the feature importance we can see that high uncertainty in prediction is associated with the regions of uncertain features (red pixels), this should indicate that the segmentation might not be correct.

Figure 4.5(a) is a good example to detect false negative predictions. We can see in the first picture from the left that no tumor was detected by the segmentation algorithm, however, the uncertainty map shows some uncertain predictions which also reflects in PET and CT feature importance maps that model is highly unsure in that particular region.

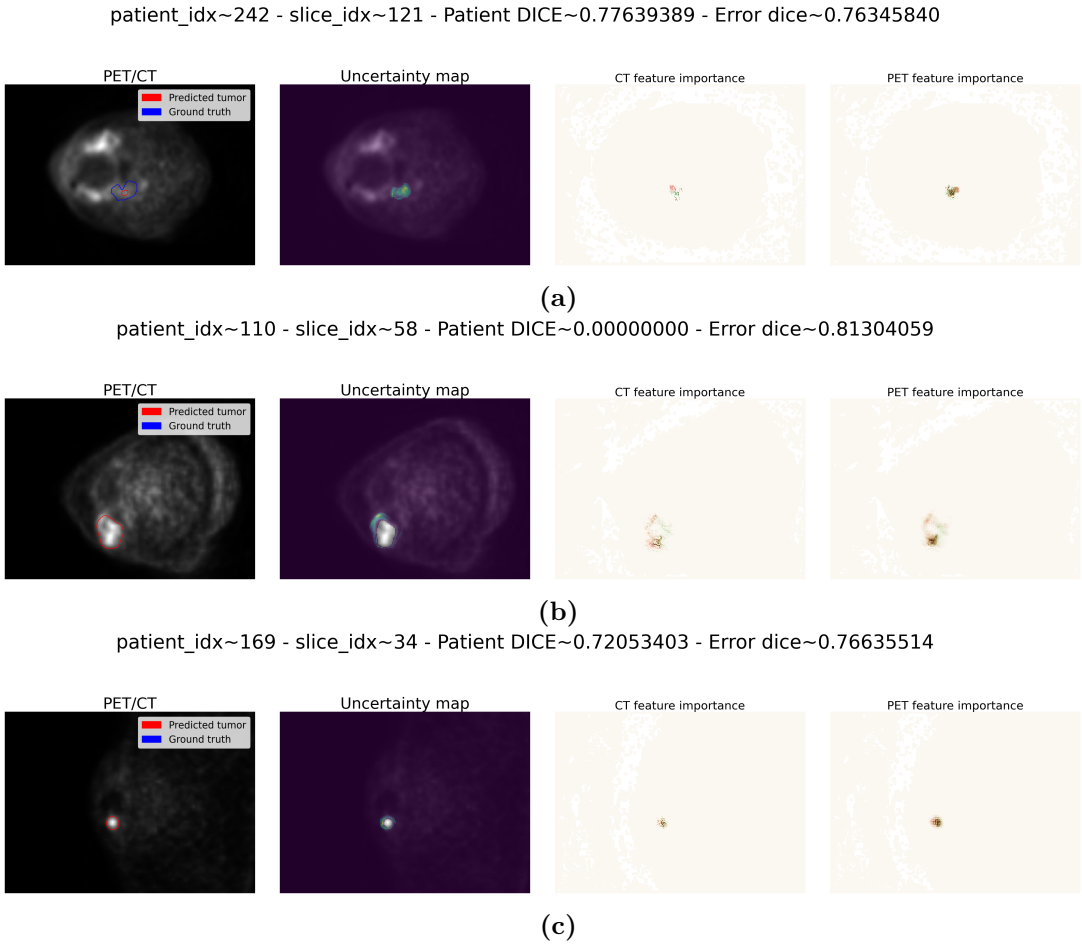


Figure 4.4: Visualization of feature importance and uncertainty in feature selection from each of the scan modalities (PET/CT). The first column on left shows PET/CT scan with prediction and ground truth contours, the second depicts the uncertainty map, thirst column from the left shows feature importance from CT scan where voxels that were deemed important by the model for segmentation are colored in green while red shows voxel where the model was uncertain of their importance. The last column depicts the feature importance of PET scans with the same color scheme as used for the CT channels. Graphics from patient_idx 242 are shown in (a) from 110 in(b) and from patient_idx 169 in (c)

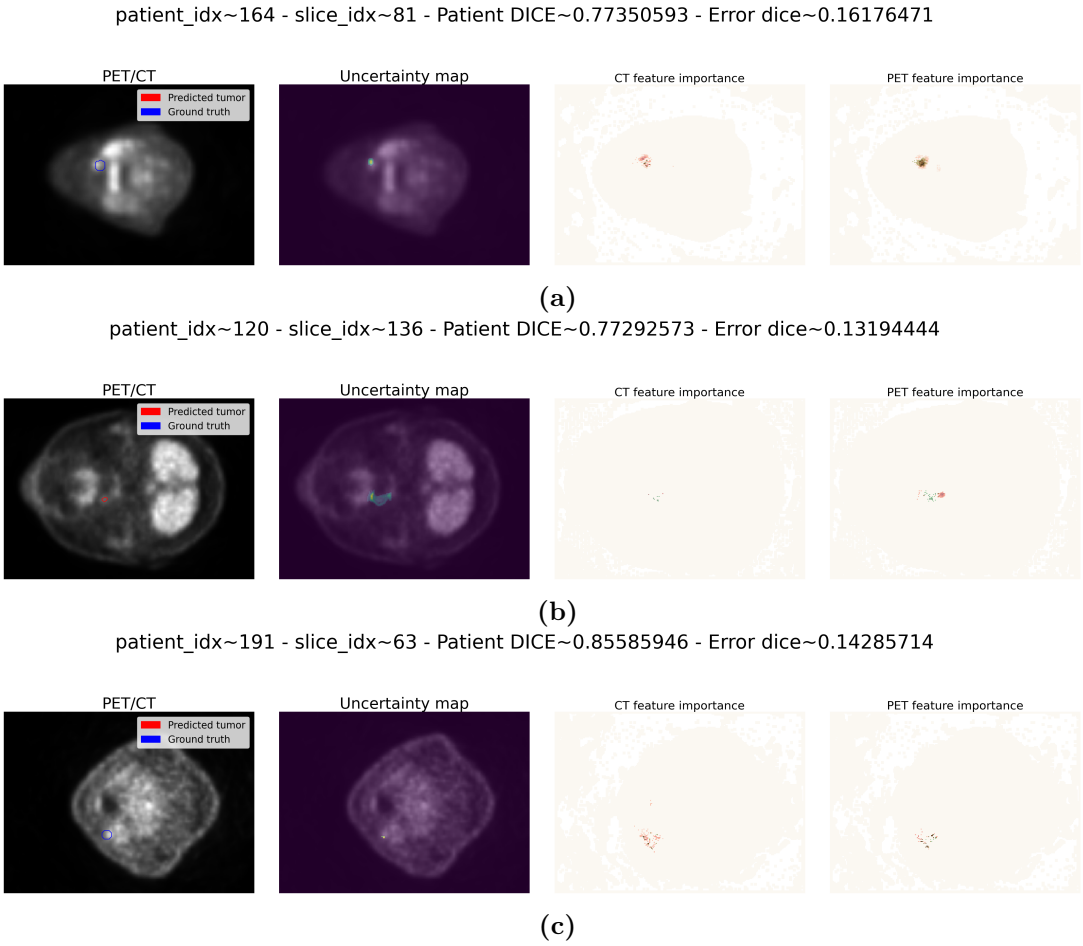


Figure 4.5: Visualization of uncertainty in feature importance for slices with low overlap between error region and uncertainty map ($dice < 0.2$). Column description is the same as explained in Figure 4.4. Here (a) shows results from patient_idx 164, (b) from patient_idx 120 and (c) from patient_idx 191 as explained in Table 4.3

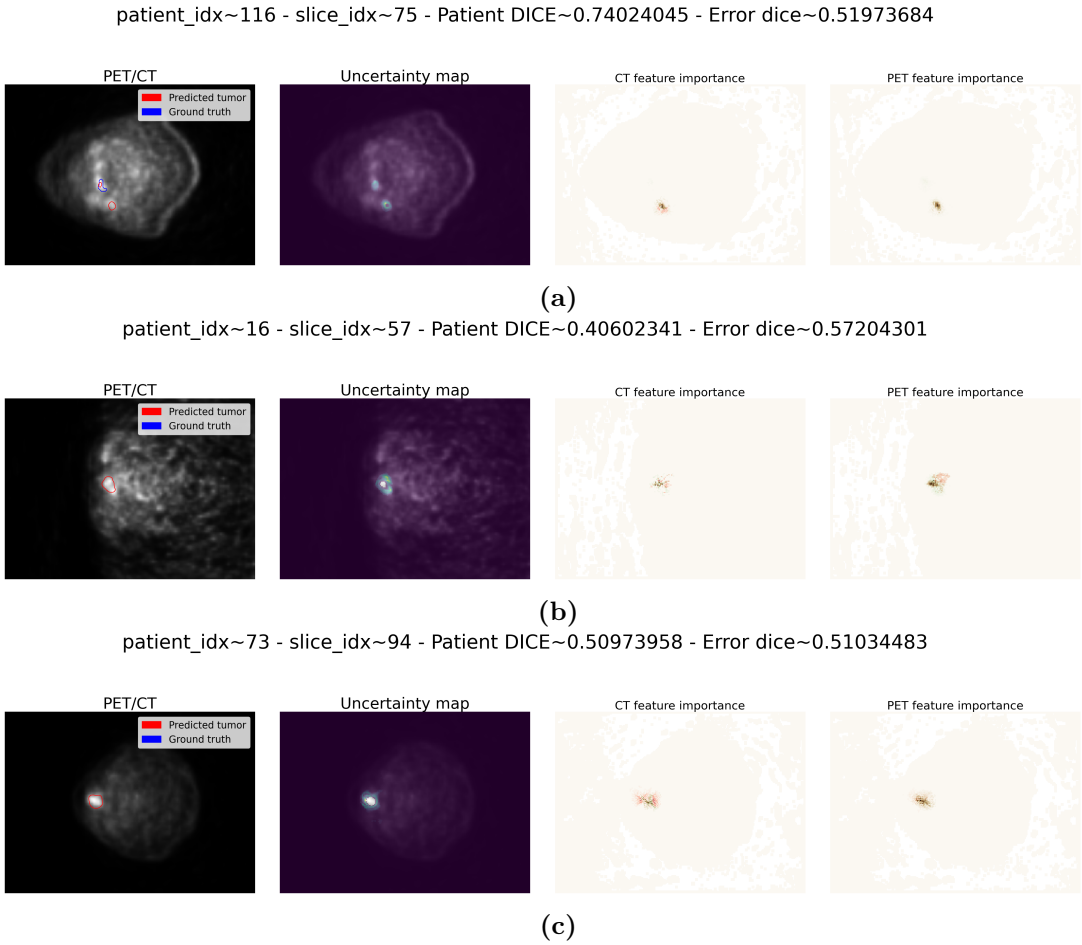


Figure 4.6: Repetition of visualization from Figure 4.4 for patient with intermediate overlap between error region and uncertainty map ($dice = 0.5 - 0.6$) as mentioned in Table 4.4

One thing that is common in all the above images is that the combination of uncertainty in feature importance and uncertainty in prediction is able to identify and highlight false predictions made by convolutional neural networks and also true positive predictions where the model was really sure. This potentially gives clinicians enough additional information to interpret the segmentation made by the model.

4.3 Uncertainty in prediction

4.3.1 Qualitative analysis

Chapter 3 explained the approach of monte carlo dropouts and its implementation for this thesis that was used as a Bayesian approximation. The mean from the distribution of predictions was used as the final segmentation and the standard deviation which was thresholded using mean + 3 * (standard deviations) was used to produce uncertainty map. In this section were are performing a qualitative analysis of results from uncertainty maps and how they were able to define false predictions in the model.

Error region is defined as the region in a scan that was predicted incorrectly by the deep learning model. This included both false positive and false negative predictions. To quantify if uncertainty maps which are obtained as Bayesian approximation are able to highlight false predictions in the model, we employed an overlap metric of uncertainty map and error region in the test dataset, measured in dice score. Table 4.5 below summarizes the results when uncertainty maps were obtained using model.2 and model.3 in the test cycle as defined in Table 3.2. An average dice score of 0.228 and 0.211 was achieved by model.2 and model.3 respectively. 'Max. dice' shows the maximum overlap between uncertainty and error region which was defined by each of the models.

Table 4.5: Model performance measured in the overlap between error region and uncertainty map

Model	Learning rate	Dropout rate	Avg. dice	Max. dice
model.2	0.0001	0.5	0.228	0.536
model.3	0.0001	0.1	0.211	0.414

The results from the table above (Table 4.5) do not give any conclusive results on whether false predictions were successfully highlighted by uncertainty maps or not. Hence we performed some quantitative analysis using patient groups that were defined in Table 4.2, Table 4.3 and Table 4.4.

Figure 4.7, Figure 4.8 and Figure 4.9 below shows PET/CT scans with segmentation and ground truth contours as marked in plot legend in the first column

from the left, the second column highlights the error region, which is the pixel-wise difference of segmentation and ground truth and the last column visualizes uncertainty maps obtained from model.2 in the test cycle.

From Figure 4.7(a) we can see that the uncertainty map from the model highlights almost the whole segmentation and ground truth region however, uncertainty is stronger (bright voxels) just outside the predicted tumor, hence indicating a probable false negative region. The same pattern can be seen in Figure 4.9(a) where false negative predictions have higher uncertainty values associated.

Figure 4.7(b) is an example where the uncertainty map was able to highlight false positive predictions, where the uncertainty map covers almost all of the predicted segmentation and is particularly bright on the edges of the structure.

Although the quantitative stats mentioned in Table 4.5 indicated an insufficient correlation between the uncertainty map and error region, but from qualitative analysis, all these figures reflect that the uncertainty map, especially the bright region was able to highlight the error region more or less in some way. And it is expected behavior as per theory, that the uncertainty region could not fully define the error region but gives enough evidence to be interpreted as false predictions, especially where uncertainty is high.

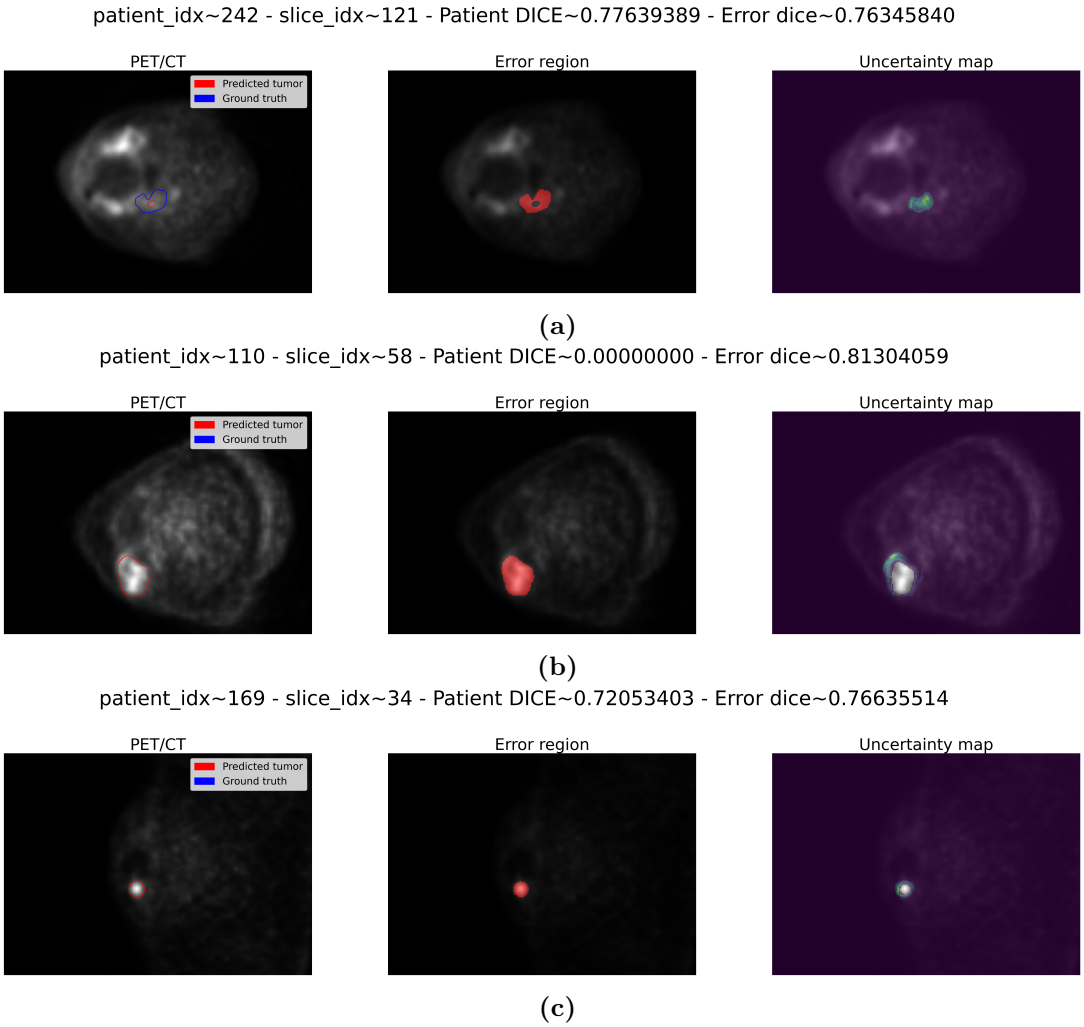


Figure 4.7: Visualization of Uncertainty in prediction for a patient with high overlap between error region and uncertainty map ($dice > 0.75$). The first column from the right shows scan with both modalities (PET/CT) plotted on top of each other and a contour drawn against ground truth (blue) and predicted tumor(red). The Middle column shows the error region for prediction, this includes both false positive and false negative predictions. The last column on right shows the uncertainty map. Patient_idx 121 is shown in (a), Patient_idx 110 in (b) and Patient_idx 169 in (c)

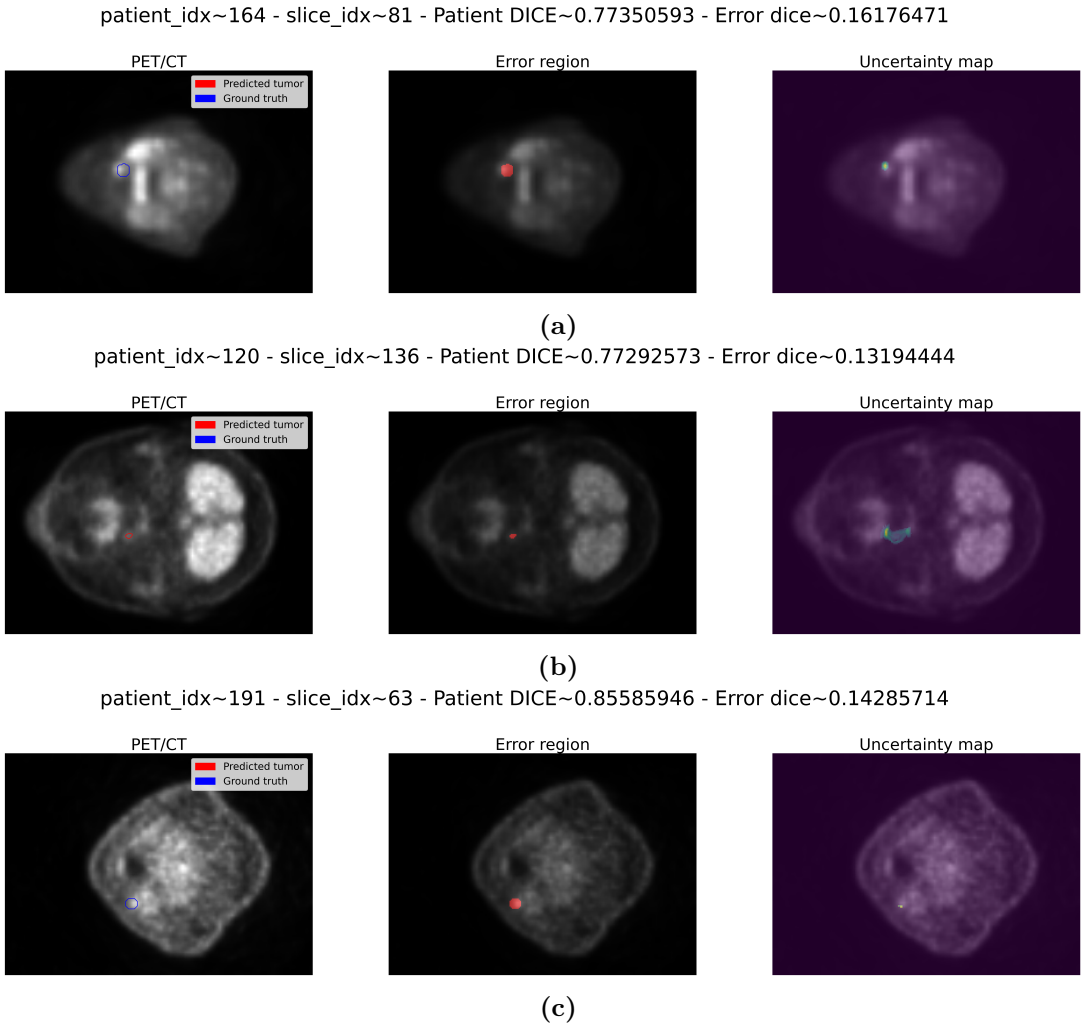


Figure 4.8: Visualization of Uncertainty in prediction for a patient with low overlap between error region and uncertainty map ($dice < 0.2$). Column descriptions are the same as in Figure 4.7. (a) visualizes results from Patient_idx 164 with error dice 0.161 (b) from Patient_idx 120 with error dice 0.131 and (c) Patient_idx 191 with error dice 0.142

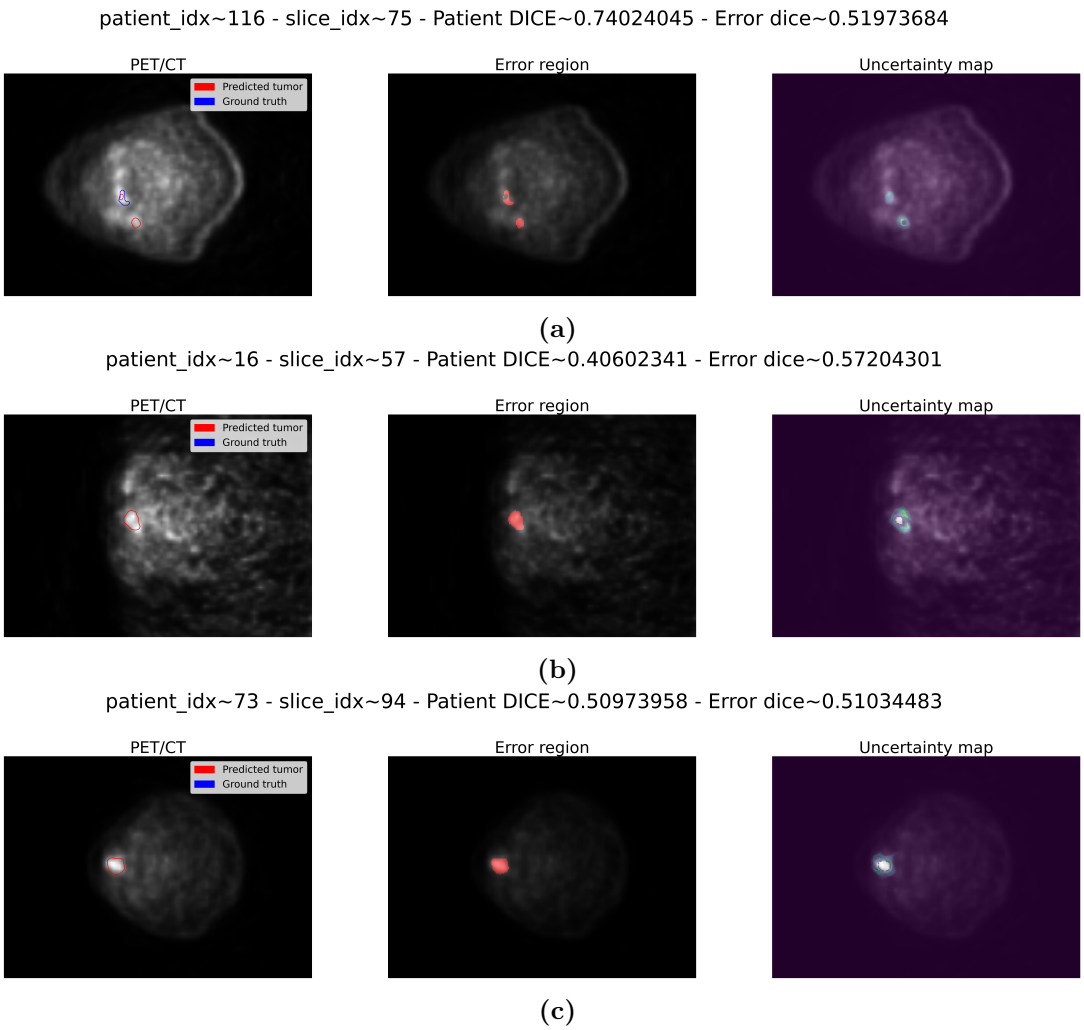


Figure 4.9: Visualization of Uncertainty in prediction for patient with intermediate overlap between error region and uncertainty map ($dice = 0.5 - 0.6$). (a)(b)(c) shows results from patient_idx 116, 16 and 73 respectively

4.3.2 Quantitative analysis

As explained in Section 3.4.2, in order to potentially improve the acceptability of deep learning models for auto segmentation of tumors in clinical settings, in addition to the visualization of feature importance and uncertainty in prediction, it is crucial to provide an estimate of confidence of model in its prediction in terms of easily interpretable and comparable metric. For this at first, we trained a simple ordinary least square (OLS) model based on average prediction uncertainty

values for each scan to predict segmentation dice score. This model was trained on the test dataset, however, it didn't perform quite well as the Figure 4.10 below shows the model could only achieve R^2 of 0.183 depicting not a good fit. This was expected as the average uncertainty value theoretically is not a comparable metric between data points.

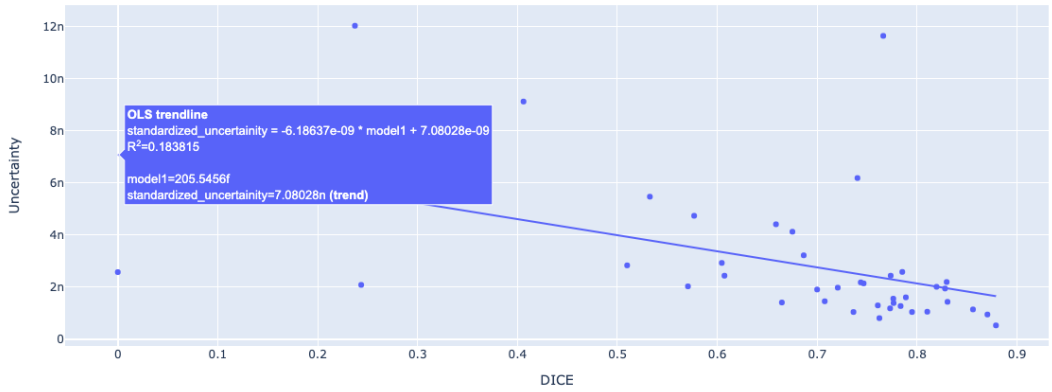


Figure 4.10: This plot illustrates the performance of an Ordinary least-squares (OLS) model when applied to predict segmentation dice score from average uncertainty value for scans in the test dataset. The coefficient of determination achieved for this model is 0.183

We adopted another novel approach, inspired by a study from [36] to quantify patient level segmentation confidence. For this, we used an overlap metric of uncertainty map and predicted tumor measured in dice to estimate segmentation dice. In theory, a high overlap between uncertainty map and predicted tumor should represent a low segmentation dice or vice versa, as that would mean the model is not certain in its prediction when the overlap is high.

To achieve this, four shallow learning regression models were trained to predict segmentation dice as shown in Table 4.6. Gradient boosting regressor outperformed all others with an R^2 value of 0.728 and 0.487 with uncertainties calculated through model.2 and model.3 respectively. Figure 4.11 below shows predicted dice scores plotted against actual dice score of segmentation, these predictions were made using Gradient boosting regressor, while the input variable (i.e. dice score from the overlap of uncertainty map and predicted tumor) was calculated based on uncertainty maps estimated using model.2.

Table 4.6: Regression models to predict segmentation dice score from the overlap of segmentation and uncertainty map

	$dropout = 0.5$			$dropout = 0.1$		
Regression Model	R^2	MSE	RMSE	R^2	MSE	RMSE
GradientBoostingRegressor	0.728	0.009	0.095	0.487	0.017	0.131
RandomForestRegressor	0.469	0.017	0.133	0.534	0.015	0.124
LinearRegression	0.012	0.033	0.181	0.0124	0.033	0.181
Ridge	0.023	0.032	0.180	0.0096	0.0331	0.182

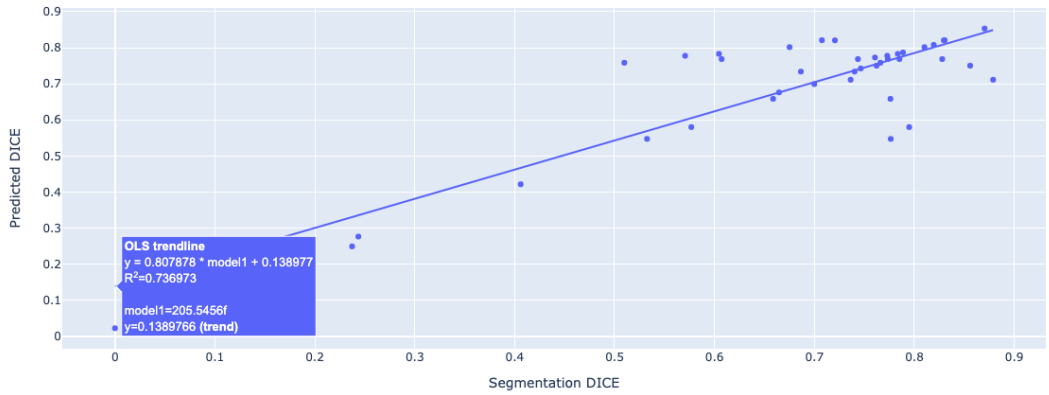


Figure 4.11: Graphs show results from our novel approach to estimate segmentation dice score based on the overlap between uncertainty map Section 3.4.2 and actual segmentation from deep learning model using Gradient boosting regression. On the horizontal axis, we have the dice score from the predicted tumor and the vertical axis shows the estimated dice score. These results were obtained by using model.2 Table 3.2 in test time to make predictions

Figure 4.12 below show repetition of the same approach using uncertainty map through model.3. In this experiment gradient boosting regressor achieved an R^2 of 0.487 and MSE of 0.017 while Random forest regressor achieved R^2 of 0.534 and MSE of 0.015 as shown in Table 4.6.

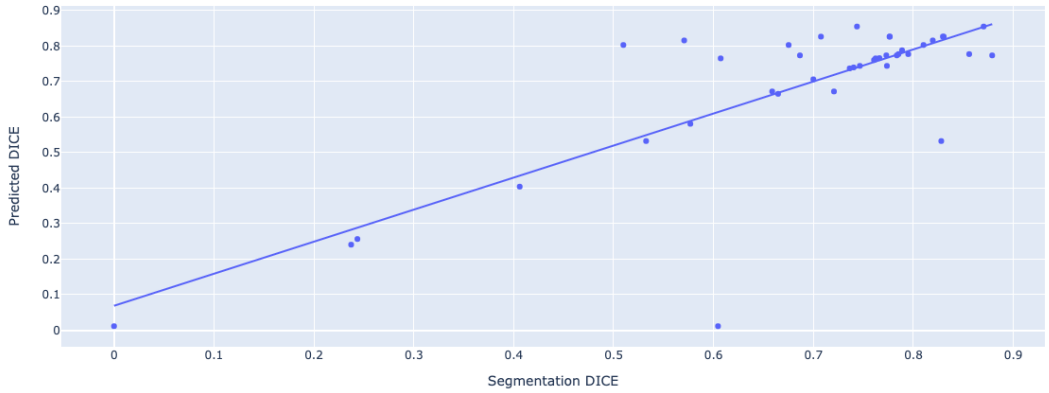


Figure 4.12: Repeated experiment as mentioned in Figure 4.11 with model.3, which had 0.1 dropout rate. Coefficient of determination R^2 using gradient boosting regression for this was calculated to be around 0.487

Chapter 5

Discussion

In this chapter, we have discussed and analyzed results obtained from the methods explained in Chapter 3 in further detail. It is important to mention here that all experiments were executed using the *deoxys* framework developed by Huynh [16] and the predictions and feature importance obtained using guided back-propagation were saved in 20 different HDF5 files. These 20 files represent the output from each iteration of monte carlo method and were analyzed separately using a different code-base (https://github.com/ahmarabbas14/MS_uncertainity) to obtain uncertainty maps and further analysis.

The results discussed in Chapter 4 concur with the theory explained in Chapter 2 and other studies [26][36] which were the inspiration for this thesis and were used as the basis for this study, as we have seen from the results that uncertainty maps obtained using Monte Carlo dropouts does explain and highlight false predictions made by convolutional neural networks.

We also did establish that using dropout layers does improve the performance of models in the training stage by regularizing models and preventing them from over-fitting. And the same dropout layers can be used for Bayesian approximation [21] using monte carlo approach.

The results obtained from different experiments are further discussed and compared in the sections to follow.

5.1 Importance of dropout layers

Figure 4.1 clearly shows that models trained with dropout layers have rather stable performance over epochs as compared to the ones without dropout layers, this shows the effect of strong regularization. However, this is done over the cost of more epochs during the training process which is expected and in line with theory.

Figure 4.3 illustrates that using dropouts not only helps in regularizing models and better performance with the validation dataset but also outperforms the non-dropout based model with the test dataset. During experiments, model variants only differ from one another based on the dropout and learning rates as shown in Table 3.2. However, it will be interesting to compare the performance of models based on different placements of the dropout layer in combination with other hyper-parameters.

Our results also comply with other studies regarding the efficacy of dropouts to improve model performance [38][39]

5.2 Qualitative and quantitative analysis of prediction uncertainty

In section Section 4.3 and Section 4.2 we briefly talked about the results which were obtained by visualizing uncertainties in predictions and input feature selection. In this section, we are discussing the interpretation of those results, whether these visualizations are sufficient to identify false prediction and segmentation confidence, and their interpretation in the clinical environment.

False predictions

Table 4.5 shows that the average dice score calculated based on the overlap between error region and uncertainty maps does not reflect significant performance, however, if we compare this theoretically, these results are expected as we have thresholded uncertainty maps to mean + 3 * (STD) for every scan and there must be uncertainty in the true positive region as well. Hence a better choice of metric in this case would have been precision as mentioned by [5]

However the qualitative analysis in Section 4.3 shows that uncertainty maps were able to highlight false predictions in almost all patient groups which were selected for analysis. Figure 4.7 (a) is the perfect example to interpret that uncertainty maps in predictions and feature importance combined show high variance in the region corresponding to false predictions.

True predictions

Figure 4.4(a) shows that the uncertainty values in the true predicted region are very low as compared to the false prediction region. Also, the model is really sure about the corresponding feature importance map as we can see only green pixels highlighted in the PET and CT importance map for true positive predictions.

5.3 Visualization of uncertainties

Based on the finding and discussion above, we find it imperative to recommend a holistic view for clinicians, that should assist them while making delineations in PET/CT scans and would potentially improve the acceptability and reliability of deep learning models in the clinical environment.

Figure 5.1 below shows a sample view for radiologist's interpretation. These visualizations were created from a patient scan in test dataset with a moderately high dice score (0.720). From left to right, we have the predicted tumor delineated over scan slice, followed by uncertainty map and feature importance map from CT and PET modalities respectively.

There are two structures predicted by the CNN, the structure predicted in the bottom part of the image has very low corresponding uncertainty values, which means the algorithm is very sure in its prediction, and the corresponding feature importance in PET and CT scans also reflect that CNN is very sure about the importance of those pixels in the input slice, hence its easier to interpret that the segmentation is highly confident in this case.

The second predicted structure in this slice, on the top side of the image has high uncertainty values associated with both uncertainty map and feature importance, reflecting that this might be a false prediction and would require manual investigation from radiologists.

The label of the image shows patient and slice IDs, followed by an estimated dice score on a scale of 0-1, the value of 0.8204 reflects that model is very certain in its prediction and should help radiologists with interpreting results.

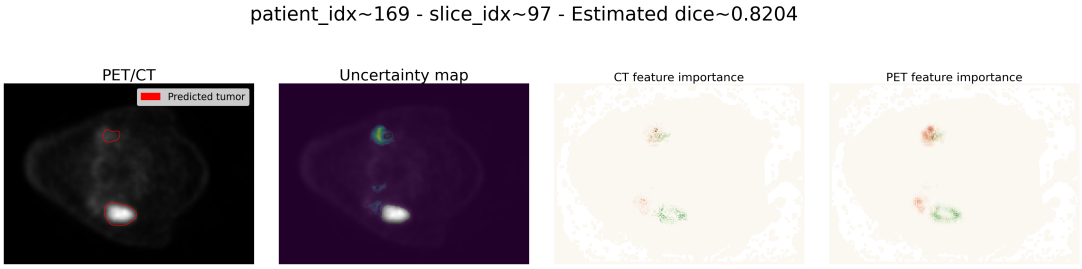


Figure 5.1: Illustration of visualization recommended to be presented to clinicians to potentially improve interpretation of auto-segmentation. The first picture from the left shows PET/CT scan with segmentation contour. The next image gives a visualization of the uncertainty map, followed by feature importance in the CT channel and feature importance in the PET channel in the last column at the right

5.4 Different approaches for uncertainty and future work

We have discussed that monte carlo dropout method is a promising approach to quantify uncertainties in segmentation made by deep learning models. However, there are other approaches as well that can be used for Bayesian approximation. [40] has done a comparison of some of the approaches in their study which includes Aleatoric Uncertainty, Ensembles, and Auxiliary Networks. Figure 5.2 below was adopted from [21] which reflects the difference between monte carlo dropouts approach and the ensemble approach. The primary difference between these approaches is that in MC dropouts, uncertainty is estimated by deliberately turning some neurons off while making predictions, forcing the model to make slightly different predictions in each iteration. While on contrary, in the ensemble approach, as the name suggests, completely different models are trained and tested separately and the variance in their predictions is used to quantify uncertainty in prediction.

The ensemble approach requires more development efforts as multiple different models are to be trained and tested and hence require more computations. Therefore, we used monte carlo approach in this thesis, however, for future work, these two approaches can be compared in a different study.

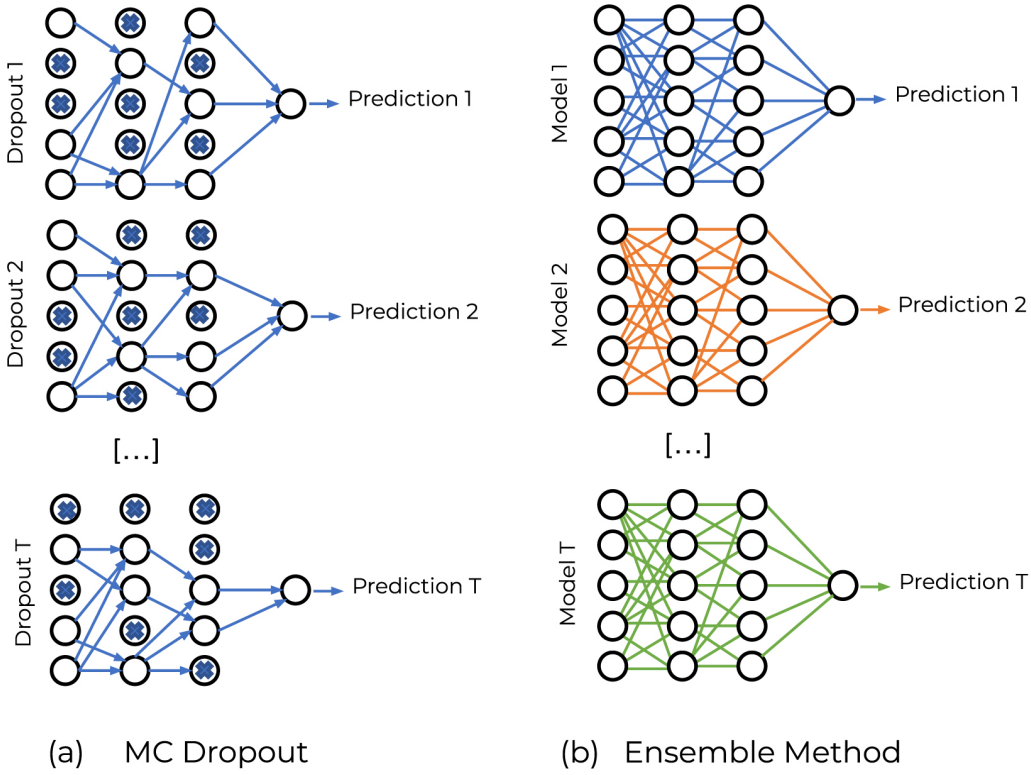


Figure 5.2: The figure shows (a) Monte carlo dropout based T predictions which were obtained by using dropout layers in test time (b) shows Ensemble method based T predictions. Adapted from [21], permitted usage under Creative Commons license

In addition to comparing monte carlo dropouts and ensemble approaches, it is also important to repeat experiments performed in this thesis with dropout layers and other hyper-parameters at different positions in the model to gauge the effect of those on uncertainty quantification.

Chapter 6

Conclusion

To summarize, the central idea behind this thesis work was to analyze the effectiveness of the dropout layer in the deep learning model and to examine if they can be used in tandem with monte carlo approach to estimate uncertainty in model predictions. For this, we performed experiments to get 20 different predictions for input scan images using different dropout models, and the variance in those predictions were used to approximate uncertainty in model predictions. In addition, we also employed guided back-propagation to approximate uncertainties in input feature selection to improve the interpretation of convolutional network models. Results from our experiments go in line with other studies on the same subject [26][36][21][5] as discussed in the section before. Contributions of our work are highlighted in Section 6.1

6.1 Contributions of our work

Based on all the experiments that were performed and the analyses which were made on the outcomes of those experiments to gauge the effectiveness of using monte carlo dropouts method, in this thesis, we were able to contribute the following:

- Firstly our quantitative analysis shows that the use of dropout layers in convolutional neural network models helps with regularization and improves its performance on test images. And these dropout layers can be kept active at the time of making predictions to estimate uncertainty in the model

predictions.

- We exhibited that monte carlo dropout method is an effective method for Bayesian approximation in a deep learning model. Our proposed uncertainty visualizations obtained from this method can potentially help clinicians at the time of their assessments and subsequently improve reliance on CNNs for auto tumor segmentation.

- We experimented and showed outcomes of a novel approach to quantify prediction certainty as inspired from [36] and achieved significant accuracy in predictions.

Bibliography

[1] WHO, *Cancer*, Feb. 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cancer.

[2] Statista, *Forecasted number of deaths from cancer worldwide from 2020 to 2040*, May 2022. [Online]. Available: https://www.statista.com/statistics/1031323/cancer-deaths-forecast-worldwide/.

[3] D. G. Pfister, S. Spencer, D. Adelstein *et al.*, 'Head and Neck Cancers, Version 2.2020, NCCN Clinical Practice Guidelines in Oncology,' *Journal of the National Comprehensive Cancer Network*, vol. 18, no. 7, pp. 873–898, 2020. DOI: 10.6004/jnccn.2020.0031.

[4] *Head and Neck Cancers*, May 2021. [Online]. Available: https://www.cancer.gov/types/head-and-neck/head-neck-fact-sheet.

[5] R. Camarasa, D. Bos, J. Hendrikse *et al.*, 'Quantitative Comparison of Monte-Carlo Dropout Uncertainty Measures for Multi-class Segmentation,' *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis*, pp. 32–41, 2020. DOI: 10.1007/978-3-030-60365-6\{_\}4.

[6] A. R. Groendahl, I. Skjei Knudtsen, B. N. Huynh *et al.*, 'A comparison of methods for fully automatic segmentation of tumors and involved nodes in PET/CT of head and neck cancers,' *Physics in Medicine & Biology*, vol. 66, no. 6, p. 065 012, 2021. DOI: 10.1088/1361-6560/abe553.

[7] Y. M. Moe, 'Deep learning for automatic delineation of tumours from PET/CT images,' Tech. Rep. Masters thesis, 2019.

[8] A. Jijja and D. Dinesh, 'Efficient MRI Segmentation and Detection of Brain Tumor using Convolutional Neural Network,' *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 4, 2019. DOI: 10.14569/ijacsa.2019.0100466.

[9] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, 'Gradient-based learning applied to document recognition,' *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. DOI: 10.1109/5.726791.

[10] T. Magadza and S. Viriri, 'Deep Learning for Brain Tumor Segmentation: A Survey of State-of-the-Art,' *Journal of Imaging*, vol. 7, no. 2, p. 19, 2021. DOI: 10.3390/jimaging7020019.

[11] H. Jiang, Z. Diao and Y.-D. Yao, 'Deep learning techniques for tumor segmentation: a review,' *The Journal of Supercomputing*, vol. 78, no. 2, pp. 1807–1851, 2021. DOI: 10.1007/s11227-021-03901-6.

[12] *NCBI - WWW Error Blocked Diagnostic*, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK279410/.

[13] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.

[14] S. Albawi, T. A. Mohammed and S. Al-Zawi, 'Understanding of a convolutional neural network,' *2017 International Conference on Engineering and Technology (ICET)*, 2017. DOI: 10.1109/icengtechnol.2017.8308186.

[15] I. C. Education, *Convolutional Neural Networks*, Jan. 2021. [Online]. Available: https://www.ibm.com/cloud/learn/convolutional-neural-networks.

[16] B. N. Huynh, 'Visualization of Deep Learning in Auto-Delineation of Cancer Tumors,' Tech. Rep. Master's thesis, 2020.

[17] Y. Liu, '3D Image Segmentation of MRI Prostate Based on a Pytorch Implementation of V-Net,' *Journal of Physics: Conference Series*, vol. 1549, no. 4, p. 042 074, 2020. DOI: 10.1088/1742-6596/1549/4/042074.

[18] T. Bayes, 'Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s,' *Philosophical transactions of the Royal Society of London*, no. 53, pp. 370–418, 1763.

[19] P. Afshar, A. Mohammadi and K. N. Plataniotis, 'Bayescap: A bayesian approach to brain tumor classification using capsule networks,' *IEEE Signal Processing Letters*, vol. 27, pp. 2024–2028, 2020.

[20] Z. U. Abideen, M. Ghafoor, K. Munir *et al.*, 'Uncertainty assisted robust tuberculosis identification with bayesian convolutional neural networks,' *Ieee Access*, vol. 8, pp. 22 812–22 825, 2020.

[21] A. Barragán-Montero, A. Bibal, M. H. Dastarac *et al.*, 'Towards a safe and efficient clinical implementation of machine learning in radiation oncology by exploring model interpretability, explainability and data-model dependency,' *Physics in Medicine & Biology*, vol. 67, no. 11, 11TR01, 2022. DOI: 10.1088/1361-6560/ac678a.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, 'Dropout: A simple way to prevent neural networks from overfitting,' *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[23] Y. Gal and Z. Ghahramani, 'Dropout as a bayesian approximation: Representing model uncertainty in deep learning,' in *international conference on machine learning*, PMLR, 2016, pp. 1050–1059.

[24] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller and W. Samek, 'On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,' *PloS one*, vol. 10, no. 7, e0130140, 2015.

[25] J. T. Springenberg, A. Dosovitskiy, T. Brox and M. Riedmiller, 'Striving for simplicity: The all convolutional net,' *arXiv preprint arXiv:1412.6806*, 2014.

[26] K. Wickstrøm, M. Kampffmeyer and R. Jenssen, 'Uncertainty and interpretability in convolutional neural networks for semantic segmentation of colorectal polyps,' *Medical Image Analysis*, vol. 60, p. 101 619, 2020. DOI: 10.1016/j.media.2019.101619.

[27] Z. Zhu, C. Liu, D. Yang, A. Yuille and D. Xu, 'V-NAS: Neural Architecture Search for Volumetric Medical Image Segmentation,' *2019 International Conference on 3D Vision (3DV)*, 2019. DOI: 10.1109/3dv.2019.00035.

[28] F. Milletari, N. Navab and S.-A. Ahmadi, 'V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation,' *2016 Fourth International Conference on 3D Vision (3DV)*, 2016. DOI: 10.1109/3dv.2016.79.

[29] S. Lu, J. Han, J. Li, L. Zhu, J. Jiang and S. Tang, 'Three-dimensional Medical Image Segmentation with SE-VNet Neural Networks,' *2021 3rd International Conference on Intelligent Medicine and Image Processing*, 2021. DOI: 10.1145/3468945.3468948.

[30] X. Guan, G. Yang, J. Ye *et al.*, '3D AGSE-VNet: an automatic brain tumor MRI data segmentation framework,' *BMC Medical Imaging*, vol. 22, no. 1, 2022. DOI: 10.1186/s12880-021-00728-8.

[31] J. Bertels, T. Eelbode, M. Berman *et al.*, 'Optimizing the Dice Score and Jaccard Index for Medical Image Segmentation: Theory and Practice,' *Lecture Notes in Computer Science*, pp. 92–100, 2019. DOI: 10.1007/978-3-030-32245-8\{_}11.

[32] NMBU, *NMBU Orion HPC Cluster — NMBU-SUPPORT*, Nov. 2020. [Online]. Available: https://support.nmbu.no/it-dokumentasjon/nmbu-orion-regneklynge/.

[33] S. Park and N. Kwak, 'Analysis on the Dropout Effect in Convolutional Neural Networks,' *Computer Vision – ACCV 2016*, pp. 189–204, 2017. DOI: 10.1007/978-3-319-54184-6\{_}12.

[34] *tf.keras.layers.Dropout — TensorFlow Core v2.9.1*, Jul. 2022. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout.

[35] *matplotlib.pyplot — Matplotlib 3.5.2 documentation*, 2022. [Online]. Available: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html.

[36] J. Ren, J. Teuwen, J. Nijkamp *et al.*, 'OC-0771 Uncertainty map for error prediction in deep learning-based head and neck tumor auto-segmentation,' *Radiotherapy and Oncology*, vol. 170, S688–S689, 2022. DOI: 10.1016/s0167-8140(22)02677-9.

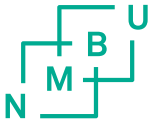
[37] A. G. Roy, S. Conjeti, N. Navab and C. Wachinger, 'Bayesian QuickNAT: Model uncertainty in deep whole-brain segmentation for structure-wise quality control,' *NeuroImage*, vol. 195, pp. 11–22, 2019. DOI: 10.1016/j.neuroimage.2019.03.042.

[38] C. Guo, M. Szemenyei, Y. Pei, Y. Yi and W. Zhou, 'SD-Unet: A Structured Dropout U-Net for Retinal Vessel Segmentation,' *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE)*, 2019. DOI: 10.1109/bibe.2019.00085.

[39] S. H. Khan, M. Hayat and F. Porikli, 'Regularization of deep neural networks with spectral dropout,' *Neural Networks*, vol. 110, pp. 82–90, 2019. DOI: 10.1016/j.neunet.2018.09.009.

[40] A. Jungo, F. Balsiger and M. Reyes, 'Analyzing the Quality and Challenges of Uncertainty Estimations for Brain Tumor Segmentation,' *Frontiers in Neuroscience*, vol. 14, 2020. DOI: 10.3389/fnins.2020.00282.

Thank you.



Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway