Norwegian University
of Life Sciences

Master's Thesis 2024 30 ECTS
Faculty of Science and Technology

Survival Analysis Using Deep Learning Models for Head and Neck Cancer Patients

Torjus Strandenes Moen
Data Science

Acknowledgments

I would like to thank my supervisor Professor Cecilia Marie Futsæther for her support during the writing of this thesis. Her guidance and feedback were invaluable to the research conducted.

A special thanks to Ph.D. candidate Bao Ngoc Huynh for making the framework used in this thesis, and for always answering any questions and providing assistance.

Additionally, I would like to thank the friends I made during my time at NMBU, especially Vetle Aasen Reinholt, Kim Næss Kynningsrud and Ulrik Egge Husby for their solidarity during this semester. I am thankful for the memorable moments we shared and for their loving companionship.

Aut frangitur aut sustinet!

Torjus Strandenes Moen
Ås, May 2024

Abstract

This thesis made deep learning models for time-to-event prediction on patients with head and neck cancer. The follow-up period was split into ten time intervals, spanning a five year period. The models predicted the time until two different endpoints, overall survival and disease free survival. Models, based on the EfficientNet architecture, were given combinations of CT and PET images, and contours around the primary tumor and nodal areas, as input. The models were evaluated using Harrell's Concordance Index (C-index), the Area Under the Receiver Operating Characteristic Curve (AUC) and the Integrated Brier Score (IBS) as metrics, using one internal dataset to train and validate the models, and one external dataset for testing. The model that achieved the highest overall performance utilized CT, PET, and a primary tumor contour as inputs, and was able to achieve a C-index of 0.74, AUC of 0.69 and IBS of 0.16 on the internal dataset. The predictions on the overall survival endpoint were generally of a higher score than predictions on the disease free survival endpoint, across all metrics.

Additionally, the models were assessed on their explainability, detailing how the model predictions related to the observed real data, and what parts of the images the models used for predictions. This was done using Kaplan-Meier curves, and saliency maps from the Variance of the Model Gradients method and the Shapley Additive Explanations method. The Kaplan-Meier curves indicated that the models generally overestimated the survival probabilities of all patients. Saliency maps generated using the Variance of the Model Gradients method and the Shapley Additive Explanations method showed that the PET modality was the most influential in model predictions, while the CT modality had the least influence. The models mostly took information from the primary tumor area when predicting on the overall survival endpoint, and from both the primary tumor and nodal areas when predicting on the disease free survival endpoint. The Shapley Additive Explanations method for explaining the model predictions proved to show the same areas as the Variance of the Model Gradients method.

Table of Contents

List of Abbreviations

AUC Area Under the Receiver Operating Characteristic Curve

BS Brier Score

BS(t) Time Dependent Brier Score

BS^c(t) Time Dependent Brier Score Under Random Censorship

C-index Harrel's Concordance Index

CNN Convolutional Neural Network

CT Computed Tomography

DFS Disease Free Survival

DM Distant Metastasis

FDG Fluorodeoxyglucose

FPR False Positive Rate

GTV Gross Tumor Volume

HECKTOR HEad and neCK TumOR

HNC Head and Neck Cancer

HPV Human Papillomavirus

HU Hounsfield Units

IBS Integrated Brier Score

KM Kaplan-Meier

LIME Local Interpretable Model-agnostic Explanations

MAASTRO Maastro Clinic Maastricht

OS Overall Survival

OUS Oslo University Hospital

PET Positron Emission Tomography

ROC Receiver Operating Characteristic Curve

SHAP SHapley Additive exPlanations

SUV Standardized Uptake Values

TPR True Positive Rate

VarGrad Variance of the Model Gradients

Chapter 1

Introduction

1.1 Head and Neck Cancer

In Norway, there were 38 094 new cases of cancer in 2023, following 11 451 cancer-related deaths in 2022 [1].

Head and Neck Cancer (HNC) is a group of cancers in the tissues and organs of the head and neck region [2] [3]. Tumors arise in the oral and nasal cavity, pharynx, larynx, sinuses and salivary glands. Worldwide, HNC is the seventh most common cancer [4]. The most common type of HNC, accounting for over 90% of cases, is squamous cell carcinoma, and results in approximately 400 000 worldwide deaths annually [2].

Consumption of alcohol and tobacco are leading contributors to HNC due to the carcinogens they introduce to the lining of the aerodigestive pathway [5]. This increases the likelihood of developing both initial and additional cancers in the head, neck, lungs, throat, and other regions with similar risk profiles. Approximately 75% of HNC is caused by alcohol and tobacco use, while the remaining 25% is mainly caused by Human Papillomavirus (HPV) [6] [7].

Various imaging techniques are employed to determine the stage of the cancer and to develop a treatment strategy. Computed Tomography (CT) images provide a detailed view of bones, blood vessels and soft tissues, useful for cancer staging and treatment planning [8] [9]. Positron Emission Tomography (PET) images show the metabolic activity in tissues, aiding in detecting cancerous areas [10]. In addition to medical images, a Gross Tumor Volume (GTV) contour around the primary tumor (GTVp) and nodal area (GTVn) are manually delineated by oncologists, guided by the medical images [11] [12]. During the planning of cancer treatments, GTV contours are drawn around tumors and nodes to focus the treatment precisely, ensuring that healthy tissue is preserved while the tumor is targeted.

The treatment of HNC varies based on how developed the cancer is, where the cancer is located and whether or not surgery is possible [13]. Treating the cancer often includes surgery, radiation and chemotherapy.

1.2 Objectives

Radiomics involves extracting large amounts of features from medical images in a designated region of interest [14]. This process involves manual or semi-automatic feature extraction, and leads to variability and highly correlated features [15]. These issues can lead to models with poor generalizability due to the high number of features, and arbitrarily chosen features importance due to the high correlation. A Convolutional Neural Network (CNN) model can offer an alternative way of extracting features, done automatically by the CNN model from the medical images, potentially leading to better feature extraction with more robust models.

The primary objective of this thesis is to develop time-to-event CNN survival models. The models will be using HNC datasets, predicting the survival time until the endpoints Overall Survival (OS) and Disease Free Survival (DFS). One dataset will be used for model training and internal validation, and a separate external dataset, not seen by the models during the training phase, will be used for model evaluation, to ensure unbiased performance metrics. The models will be given combinations of CT, PET and GTV contours as input. This is done to automate the process of feature extraction from medical images. Crucial to this objective is assessing the different possible combinations of modalities and how they impact the model performance.

Secondly, an analysis of model interpretability is done to validate the predictions and facilitate trust and transparency in the models. The interpretability analysis is done with Kaplan-Meier (KM) curves, and saliency maps from the Variance of the Model Gradients (VarGrad) method and the SHapley Additive exPlanations (SHAP) method. These methods aim to clarify how the model performed relative to the observed data, and which features are relevant for time-to-event predictions.

1.3 Related Works

Among the contributions within the domain of time-to-event analysis, done on HNC with the use of CNNs, is the study conducted by Wang et al. [16], which explores the potential of deep learning models to predict Distant Metastasis (DM) and OS in patients with HNC. Wang et al. used a 3D-Resnet architecture combined with a time-to-event outcome model. A log likelihood loss function was used to incorporate censoring information in the model. Their study compared five different models based on PET images, CT images and/or a GTV contour as input. Evaluation of these models' predictive performance was done with Harrel's Concordance Index (C-index) and KM curves. For both the DM and OS endpoints, the PET-only model exhibited the highest C-index performance, suggesting that PET images, even without tumor and nodal volume segmentation, might be more informative for prognosis in HNC compared to CT images or combined PET+CT modalities.

The HEad and neCK TumOR (HECKTOR) challenge provides a competitive environment for researchers to test and refine their algorithms on a standardized, high-quality dataset consisting of diverse patient cases across several institutions [17] [18]. The winner of the outcome prediction task from the 2022 HECKTOR challenge, Rebaud et al. [19], developed a binary-weighted radiomics model for time-to-event analysis, predicting recurrence-free survival. The model achieved a C-index of 0.68 by using a simple nnUNet model for segmentation of the tumor, and extracting radiomics features from the segmented area.

Chapter 2

Theory

2.1 Survival Analysis

Survival analysis is a branch of statistics concerned with time-to-event data [20] [21] [22]. It addresses several limitations of conventional methods, like the handling of censored data, and considering the timing of an event. Survival data typically contain information about an event and the time until that event occurred. In the medical field, an event is commonly the death of a patient or the recurrence of a disease. The main objectives of survival analysis are to estimate and interpret survival and hazard functions, and assess the relationship of explanatory variables to survival [20].

2.1.1 Censoring

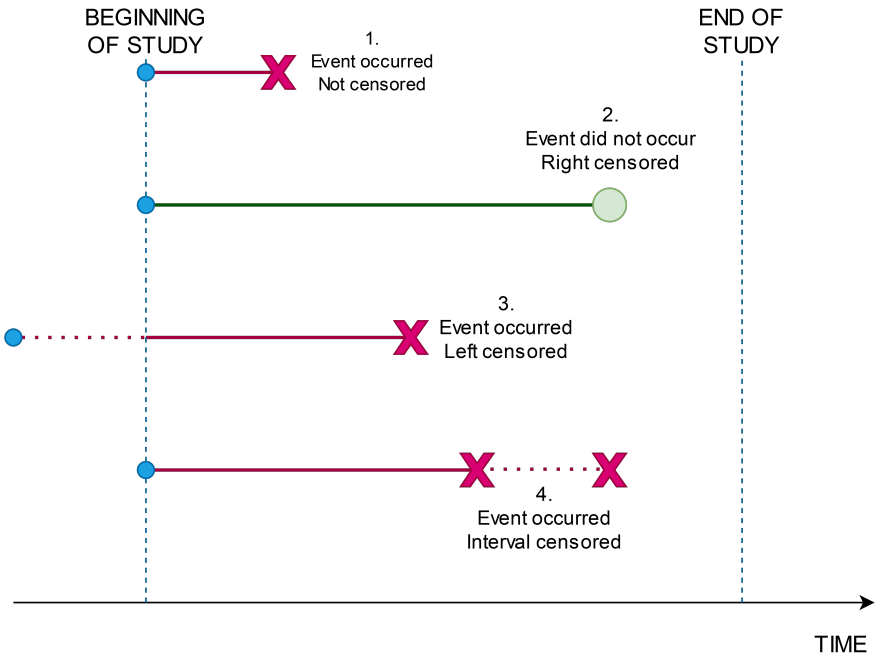


Figure 2.1: Types of censoring. 1. shows a *not censored* patient where the event occurred within the study period. 2. shows a *right-censored* patient that did not have the event occur within the study. 3. shows a *left-censored* patient whose event occurred at an unknown time before the beginning of the study. 4. shows an *interval-censored* patient whose exact event time is unknown, but known to have occurred within an interval of time.

A unique aspect of survival analysis is the handling of censored data, which occurs when there is incomplete information about the timing of the event for some study participants [21].

Reasons for incomplete information include [21]:

- The study ends before the event was observed for a patient.
- Loss to follow-up, meaning that the patient is no longer available to the study.
- The patient actively withdraws from the study.

Figure 2.1 shows four types of censoring which can happen within survival analysis.

1. Not censored:

   The event occurred within the study period, and no information is required prior to the beginning of the study [21]. All information is present, therefore the patient is not censored.

2. Right-censored:

   The study ends, or the patient leaves the study, before an event is observed [21]. The exact time of the event is unknown, but it is known to be later than the last recorded time point.

3. Left-censored:

   The event occurred before the patient entered the study [21]. The exact time of the event is unknown, but it is known to have happened before the first time point.

4. Interval-censored:

   The event is known to have occurred within a specific time interval, but the exact time is unknown [21]. This type of censoring often arises in follow-up studies where patients are checked at intervals.

Not including censoring information in the analysis would lead to inaccurate and invalid results [21]. When right-censoring information is ignored, the analysis would assume that all patients who did not have the event occur within the study had the event occur when they were censored. This leads to a bias of underestimating the survival times, since it falsely shortens the observed survival times of right-censored individuals [22]. Excluding left-censoring information leads to an underestimation of survival times, since the analysis would incorrectly assume that these individuals were at risk only from the beginning of the study period [22]. Disregarding interval-censoring leads to either underestimating or overestimating the event times, depending of where in the interval the event occurred [22]. These model biases could lead to incorrectly estimating the survival function, hazard rates, and the effects of covariates on survival times [22].

2.1.2 Survival Function

The survival function, given in Equation 2.1, is the probability that a subject's time until event, T, is greater than some specified time t [22]. That is, $S(t)$ gives the probability that an individual survives longer than a specific time t.

$$S(t) = P(T > t) \tag{2.1}$$

The survival function is non-increasing, since the probability of surviving past a certain point either decreases or stays the same as time progresses [22]. At $t = 0$ the probability of surviving is necessarily 1, and therefore $S(t = 0) = 1$. As t approaches infinity, survival must necessarily approach 0, therefore $S(t = \infty) = 0$. The value of $S(t)$ for a dataset at any given time is a direct measure of the proportion of individuals that is expected to survive beyond that time.

2.1.3 Hazard Function

The hazard function, $h(t)$, is the instantaneous rate of an event occurring at time t, given no prior occurrence up to that time [22]. Unlike the survival function, which gives the probability for surviving beyond a certain time, the hazard function gives the immediate risk of event occurrence at a certain time point. The hazard function, given in Equation 2.2, is the limit of the probability of an event occurring in the interval $[t, \quad t+\Delta t)$, divided by the length of the interval, as Δt approaches 0. Where T is the time until the event occurs.

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t} \tag{2.2}$$

The hazard function is a probability per unit of time, and therefore a rate of hazard [22]. It has no upper bound, ranging from 0 to infinity. This hazard rate is not directly interpretable, but can be interpreted by comparing it to other hazard rates.

The hazard function can be derived from the survival function [22], shown in Equation 2.3, given by

$$h(t) = -\frac{\frac{d}{dt}S(t)}{S(t)} \tag{2.3}$$

And likewise the survival function can be derived from the hazard function [22], shown in Equation 2.4, given by

$$S(t) = e^{-\int_0^t h(u)\,du} \tag{2.4}$$

This relationship shows the survival functions dependence on the cumulative hazard function [22]. The cumulative hazard function $H(t)$, shown in Equation 2.5, is a measure of the accumulated risk up to a specified time t, given by

$$H(t) = \int_0^t h(u)\,du \tag{2.5}$$

2.2 Kaplan-Meier Curves

The KM estimator is a statistic used to estimate the survival function from time-to-event data [23]. The KM estimator calculates the survival probabilities without assuming a statistical distribution for the data. The KM estimator $\hat{S}(t)$, shown in Equation 2.6, gives a step wise survival function where the survival probabilities decrease only at times when an event occurs [22]. Between the event times the survival probability remains constant.

$$\hat{S}(t) = \prod_{t_i \leq t} (1 - \frac{d_i}{n_i}) \tag{2.6}$$

Here, t_i are the times where at least one event has occurred in the set of total times t, d_i is the number of events that occurred at times t_i, and n_i is the number of individuals at risk before t_i, referred to as the risk set.

The KM estimator accounts for right-censoring by only calculating the survival estimate at times where an event occurs [22]. Patients who have not experienced an event are included in the risk set calculation up to their censoring time, but do not directly alter the survival probability estimates. Censoring information is assumed to be random and non-informative. This means that censoring does not provide any information about a patient's survival probability, and that there is no systematic reason for censoring occurring [22].

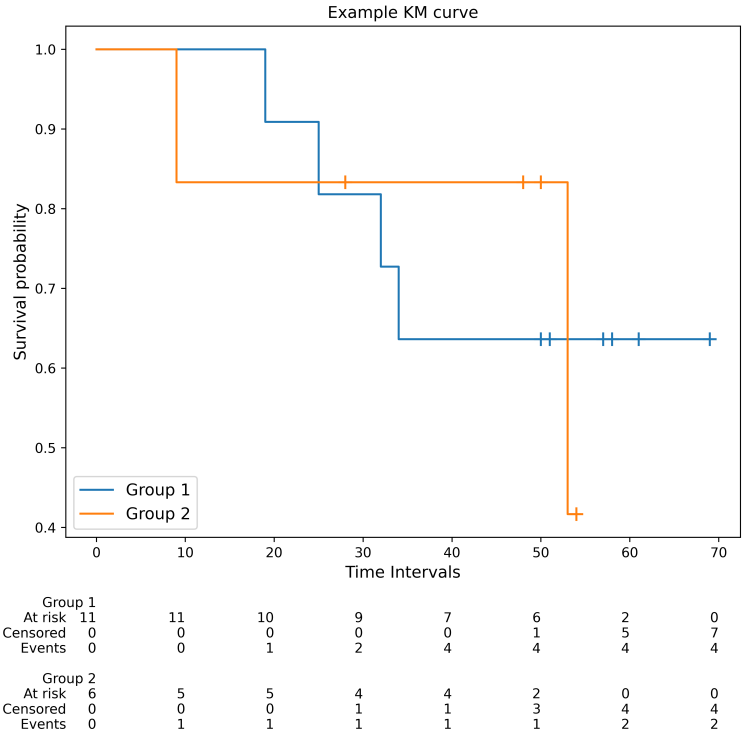


Figure 2.2: An example of KM curves, showing the estimated survival probability of two groups. Ticks on the KM curves indicates times where one or more patient were censored. Shown under the graph is a risk count. The risk count shows the number of patients that are at risk of experiencing the event, censored, or have had the event occur, at the various time intervals throughout the study.

As seen in Figure 2.2, KM curves are visual representations of the estimated survival function, $\hat{S}(t)$, over time [22]. The KM curve shows how survival probabilities decrease over time, helping identify critical periods of risk or survival benefit. KM curves can be compared to each other to uncover differences between groups, such as comparing the effect of a covariate on survival by grouping the KM curves by the covariate.

The log-rank test, first introduced as a chi-square test in [24], is used to assess statistical significant difference between two or more KM curves. The log-rank test compares the expected and observed event occurrences under the null hypothesis that there is no difference between the curves.

The log-rank statistic is calculated as follows [25] [22]:

1. For each time point an event occurs, calculate the expected number of events based on the number of individuals at risk in a group relative to the total population at risk at that time point, given in Equation 2.7, as,

$$E_{ij} = \frac{n_{ij}}{n_j} d_j \tag{2.7}$$

   where E_{ij} is the expected number of events in group i at time j. n_{ij} is the number of individuals at risk in group i just before time j. n_j is the number of individuals at risk across all groups just before time j. d_j is the total number of events observed across all groups at time j.

2. Calculate the difference, D, between the observed and expected number of events for each group at each time point, given in Equation 2.8, as,

$$D = O_{ij} - E_{ij} \tag{2.8}$$

   where O_{ij} is the number of observed events in group i at time j.

3. Calculate the variance, V_{ij}, between the observed and expected events.

4. Sum up the differences between observed and expected events, divided by the variance, to obtain the test statistic. The test statistic, χ^2, follows a chi-square distribution, as shown in Equation 2.9, where degrees of freedom equals the number of groups minus one.

$$\chi^2 = \sum_{j,i} \frac{(O_{ij} - E_{ij})^2}{V_{ij}} \tag{2.9}$$

The log-rank test assumes proportional hazards, meaning the hazard ratios between any two groups are constant over time [26] [22]. The test is still applicable if this is not the case, but it will be less likely to detect a true difference in survival between the groups. A visual way of determining if the hazard is proportional is by seeing if the KM curves cross. If they do, the proportional hazard assumption is violated. The log-rank test is said to be significant if the $p-value$ of the log-rank test is under a set threshold, usually 5%, which corresponds to a $p-value$ of 0.05.

2.3 Radiomics

Radiomics involves extracting a large number of quantitative features from a specified region of interest in images [14]. Features may be related to texture, shape and intensity of the image regions, among many others. These extracted features can be used as input for a survival model by choosing to extract features from a relevant region of interest in medical images.

Feature range in complexity from first-order, second-order and higher-ordered features [14]. First-order features are those that quantify voxels in an image without considering their spatial relationship. This includes metrics like mean or maximum voxel intensity, and distribution based features like entropy. Second-order features take spatial relationships into account, and are often texture-based features, like Gray Level Co-occurrence Matrix, which analyzes texture by measuring the frequency with which pairs of voxels with specific values occur in a specified spatial relationship within an image. Higher-order features are calculated by filters put over the image to find complex patterns. An example of a higher order feature is a wavelet transformation, which decomposes an image into multiple scales, analyzing patterns and textures at different resolutions and orientations.

A common problem with radiomics is the high number of features extracted [15]. These features are highly correlated with each other, leading to redundancy and collinearity, which complicates the model and leads to overfitting and low generalizability. One solution to this problem is the use of CNNs to automate the feature extraction and selection process [27] [28].

2.4 Convolutional Neural Networks

CNNs are a type of deep learning model specialized in processing data from multiple arrays, such as images [29]. Several types of layers are added in order to facilitate automatic feature extraction and prediction.

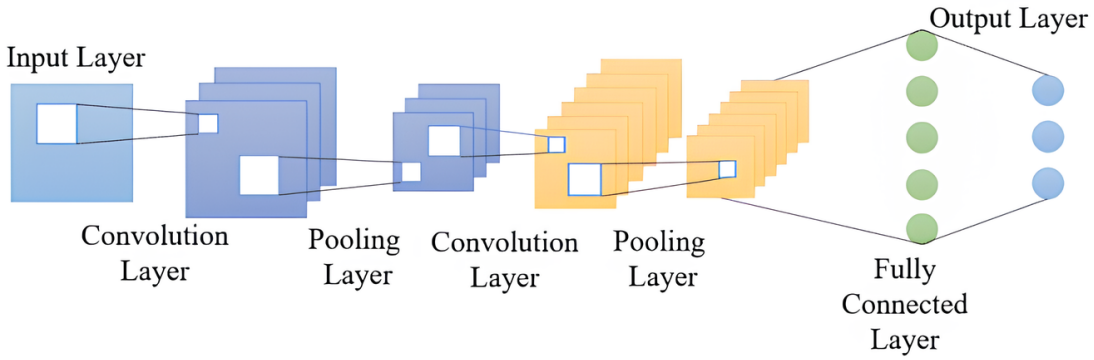


Figure 2.3: Simple CNN architecture by [30]. This work is licensed under a "CC BY 4.0" license. The image shows a simple structure of a typical CNN model, using several convolution and pooling layers after the input, and ending in a fully connected layer followed by an output layer.

As seen in Figure 2.3, the first layers of a CNN are most often convolutional and pooling layers [29]. Convolutional layers apply filters over the input and creates feature maps. These feature maps capture spatial relationships in the input, such as edges, textures, or complex patterns. At the end of a convolutional layer is an activation function that is applied to the feature map to introduce non-linearity to the model. Pooling layers are used to reduce the dimensions of the feature map, while preserving the most important information. Max pooling, the most common pooling layer, selects the maximum value from a given region.

After several iterations of convolution and pooling, a fully connected layer is added [29]. The fully connected layer connects all activations from the previous layer, and performs the higher level reasoning in the model. After the fully connected layer comes the final output layer, which outputs the models predictions.

After model predictions are made, the model error can be calculated. The loss function is a function that calculates how far the model predictions deviate from the true values [31]. Different loss functions are required for different tasks, like Mean Square Error for regression, or Cross-Entropy Loss for a classification task [29].

The CNN learns by changing the weighting between layers and in the filters in the network [29]. This is done by backpropagation, where the weights are updated relative to the gradient of the loss function, which indicates the direction to adjust the weights to minimize the error in predictions [31]. The weights are updated by an optimizer, which updates the weights to minimize the loss function [31]. The size of the updates is called the learning-rate. Different optimizers change the weights in different ways, like the Stochastic Gradient Descent optimizer [32], which updates the weights proportionally to the gradient, or the Adam optimizer [33], which dynamically changes the learning-rate for each weight.

The batch size is the number of samples used a single iteration of the model [31]. An epoch is a full pass through of the entire training dataset. The number of iterations in one epoch is equal to the total number of samples in the training dataset divided by the batch size. For example, if there are 1 000 samples in the training dataset and the batch size is 100, then one epoch will consist of 10 iterations. The model weights are updated each iteration using the optimizer. The process is repeated for several epochs. Choosing too few epochs can lead to not capturing the complexities of the data, called underfitting. Choosing too many epochs can lead to the model learning noise in the training data, and not generalizing well, called overfitting.

In survival analysis, CNNs can be faster and more consistent than conventional approaches to radiomics [27], outperforming models like the Cox proportional hazard model [25] [34]. However, CNNs tend to fail in unusual cases that are not represented well in the training data, and are harder to interpret due to their architectures which transform the data in ways that are not straightforward understand [35] [36].

2.5 The Negative Log Likelihood Loss Function

Gensheimer et al. [37] introduced an approach for analyzing survival data using a discrete-time framework adapted for neural networks. This method is particularly advantageous in handling the intrinsic complexities of survival data, including right-censored observations and time-varying hazards. The core of this approach lies in their formulation of the log likelihood loss function.

According to [37], the follow-up period is divided into discrete time intervals. The probability of an event occurring in a given interval j is the conditional hazard probability, denoted by h_j, assuming the individual has survived up to that interval. The survival probability until the end of interval j, represented as S_j, is defined in Equation 2.10, given by

$$S_j = \prod_{i=1}^{j}(1 - h_i) \tag{2.10}$$

The likelihood function incorporates contributions from each individual depending on their event occurrence [37]. Equation 2.11 shows the likelihood contribution, L, for an individual who experiences the event in interval j (uncensored). The likelihood contribution is the probability of surviving through interval $j-1$ multiplied by the hazard probability at interval j, h_j, and is given by

$$L_{uncensored} = h_j S_{j-1} = h_j \prod_{i=1}^{j-1}(1 - h_i) \tag{2.11}$$

The censored individual's contribution is the survival probability up to the censoring interval, as seen in Equation 2.12,

$$L_{censored} = S_{j-1} = \prod_{i=1}^{j-1}(1 - h_i) \tag{2.12}$$

To simplify the calculations, the likelihood function, L, is represented in logarithmic form, making it possible to convert the product to a summation. For a cohort of N individuals, the full log likelihood loss function, shown in Equation 2.13, is the sum of contributions from censored and uncensored individuals, combining Equation 2.11 and Equation 2.12,

$$\log L = \sum_{n=1}^{N} [\delta_n \cdot \ln(h_{n,j}) + \ln(S_{n,j-1})] \tag{2.13}$$

Here, δ_n is an indicator variable with the value 1 if the event occurs for individual n and 0 if the event does not occur.

A neural network survival model will minimize the loss function, while the log likelihood function should be maximized. Therefore, the loss function, defined in Equation 2.14, is given by the negative of the log likelihood,

$$-\log L = -\sum_{n=1}^{N} [\delta_n \cdot \ln(h_{n,j}) + \ln(S_{n,j-1})] \tag{2.14}$$

2.6 Evaluation Metrics

2.6.1 Area Under the Receiver Operating Characteristic Curve

The Area Under the Receiver Operating Characteristic Curve (AUC) is a widely used metric for evaluating the performance of binary classification models [38]. It offers a comprehensive measure of a model's ability to distinguish between two classes. In the context of survival analysis, these two classes will be whether or not the event occurred [39].

The AUC is derived from the Receiver Operating Characteristic Curve (ROC) [39]. The ROC is created by plotting two parameters, True Positive Rate (TPR) and False Positive Rate (FPR).

The TPR, shown in Equation 2.15, is the proportion of correctly predicted positive classes divided by the total number of actual positive classes, given by

$$\text{TPR} = \frac{TP}{TP + FN} \tag{2.15}$$

where TP is the number of true positive model predictions and FN is the number of false negative predictions.

The FPR, shown in Equation 2.16, is the proportion of falsely predicted positive classes, divided by the total number of actual negative classes, and is given by

$$\text{FPR} = \frac{FP}{FP + TN} \tag{2.16}$$

where FP is the number of false positive predictions and TN is the number of true negative predictions.

The model predicts a probability of each sample belonging to one of the two classes. To decide which class a probability score corresponds to, a classification threshold is used [38]. The classification threshold is a cut-off value that determines whether a probability score classifies an instance as belonging to the positive class or the negative class. If the probability score of an instance is above this threshold, the instance is classified as positive, otherwise, it is classified as negative. As the classification threshold changes from 0 to 1, the TPR and FPR will change, and plotting these changes yields the ROC curve [38], as seen in Figure 2.4.

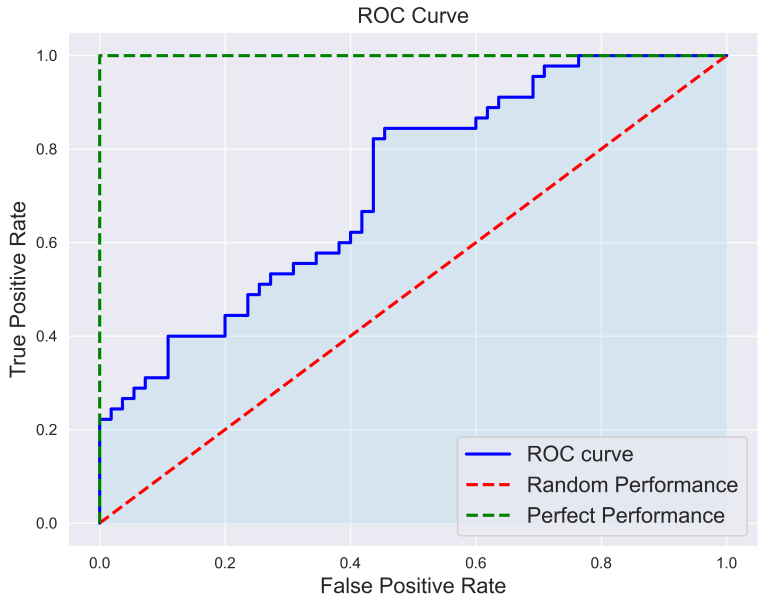


Figure 2.4: An example of ROC curves. The performance of a model making random predictions is shown with the red dashed curve, indicating half of the model predictions were false positives. Perfect performance is shown with the green dashed curve, which follows the y-axis up to 1 and follows the top, indicating zero false positive predictions. A typical ROC curve is shown with the blue curve, in between random and perfect performance. The blue curve is jagged, as opposed to smooth, because the threshold moves in discrete steps and is not continuous. The AUC corresponding to the typical ROC curve is shown in the shaded area underneath the blue curve.

The AUC quantifies the entire two-dimensional area underneath the ROC curve [38]. This way it provides a single scalar value to assess the model's ability to discriminate between positive and negative classes. The AUC value ranges from 0 to 1:

- AUC $= 1$: The model has perfect discrimination ability, correctly classifying all positive and negative instances. This corresponds to the dashed green curve in Figure 2.4.
- $0.5 <$ AUC < 1: The model has a good to excellent discrimination ability. The higher the AUC, the better the model is at correctly predicting the classes. This corresponds to the jagged blue curve in Figure 2.4.
- AUC $= 0.5$: The model has no discrimination ability and is randomly guessing. It is unable to distinguish between the two classes, as shown in the dashed red curve in Figure 2.4.
- AUC < 0.5: The model performs worse than random chance. This scenario suggests that predictions are inversely related to the actual values.

2.6.2 Harrell's Concordance Index

Harrell et al. [40] presented a method for evaluating the informational value of medical tests. The C-index measures the predictive accuracy of a model in terms of its ability to correctly rank pairs of observations. This metric extends the concepts of the AUC to accommodate censored data [17]. When dealing with binary outcomes and no censored data, the C-index is equivalent to the AUC. Both metrics then assess the model's ability to differentiate between two classes.

Two subjects are denoted as i and j, with their respective event times T_i and T_j, and predicted risk scores η_i and η_j. The metric assess whether observations with a longer survival time T are assigned a lower estimated risk score η by the model, compared to those with shorter event times [40].

A pair (i, j) is defined as [41]:

- Concordant, if the model estimated risk score is higher for individual i ($\eta_i > \eta_j$) and individual i experienced the event earlier than j ($T_i < T_j$).
- Discordant, if the model estimated risk score is higher for individual i ($\eta_i > \eta_j$) but individual i experienced the event later than j ($T_i > T_j$).

Assessing the concordance of pairs is straight forward when both individuals are uncensored, that is, both T_i and T_j are known. In this case the risk scores are compared to obtain the C-index score [41].

When one of the event times, T_i, is observed, but the other, T_j, is censored, and T_j is greater than T_i ($T_j > T_i$), it is clear that patient i experienced the event first [41]. In this case, the pair (i, j) is concordant if $\eta_i > \eta_j$, and discordant if $\eta_i < \eta_j$.

On the other hand, if T_i is observed, but the other event time T_j is censored, and T_j is less than T_i ($T_j < T_i$), the order of events is ambiguous, since it is unknown whether the event for T_j would have occurred before or after T_i if it had not been censored. These pairs are not considered in the C-index calculation [41].

If both T_i and T_j are censored, it remains uncertain who experienced the event first, or if it occurred at all. Therefore, pairs with two censored individuals are excluded from the computation [41].

The full Equation for the C-index [42] is given by

$$C = \frac{\sum_{i,j:t_i<t_j} I(\eta_i > \eta_j)\delta_i}{\sum_{i,j:t_i<t_j} \delta_i} \tag{2.17}$$

where

$$I(\eta_i > \eta_j) = \begin{cases} 1, & \text{if } \eta_i > \eta_j \\ 0, & \text{otherwise} \end{cases}$$

and

$$\delta_i = \begin{cases} 0, & \text{if individual } i \text{ is censored} \\ 1, & \text{if individual } i \text{ is uncensored} \end{cases}$$

Using Equation 2.17, a score of 1 is a perfect score, where all pairs of observations are correctly ordered by their risk scores. A score of 0.5 indicates random predictive performance with no discriminating ability.

The accuracy of the C-index can be overly optimistic or inflated when there is a high degree of censoring in the data [41]. This bias occurs as the number of comparable pairs decreases, potentially leading to an overestimation of the model's predictive performance.

2.6.3 Integrated Brier Score

The Brier Score (BS) is a measure used to assess the accuracy of probabilistic predictions [43]. The metric quantifies the accuracy of predictions by comparing the predicted probabilities of events to actual outcomes at a specific time point [44], given in Equation 2.18,

$$BS = \frac{1}{N}\sum_{i=1}^{N}(o_i - \hat{\pi}(t|X_i))^2 \tag{2.18}$$

where N is the number of predictions, o_i is the actual outcome corresponding to prediction i, and $\hat{\pi}(t|X_i)$ is the predicted probability of the event for the i^{th} prediction at time point t, given the input X. The BS ranges from 0 to 1, where 0 indicates perfect accuracy and 1 denotes the lowest possible accuracy.

For time-to-event analysis, the BS can be divided into three categories based on censoring status and the time of the event occurrence relative to the time point of BS calculation [44].

These categories are [44]:

1. Individual i had the event occur before BS calculation: $t_i \leq t, \delta_i = 1$
   Here, t_i is the event time, t is the time at which the BS is being calculated and δ_i is a binary indicator of whether an individual is censored, $\delta_i = 0$, or uncensored, $\delta_i = 1$. Category 1's contribution to the BS, following Equation 2.18, is defined as Equation 2.19,

   $$BS = (0 - \hat{\pi}(t|X_i))^2 \tag{2.19}$$

   Since the individual has had the event occur, the actual outcome, o_i, is 0.

2. Individual i has not had the event occur before BS calculation, making censoring status uncertain. In this case, $t_i > t, \delta_i = 1$ or $\delta_i = 0$.
   Category 2's contribution to the BS is given by Equation 2.20,

   $$BS = (1 - \hat{\pi}(t|X_i))^2 \tag{2.20}$$

   Since the individual has not had the event occur, the actual outcome, o_i, is 1.

3. Individual i is censored before BS calculation. In this case, $t_i \leq t, \delta_i = 0$.
   Category 3's contribution to the BS is undefined since the event status at time t is unknown.

The equations for category 1 and 2 combine into Equation 2.21 for the Time Dependent Brier Score (BS(t)) [44], given by

$$BS(t) = \frac{1}{N}\sum_{i=1}^{N} I(t_i \leq t, \delta_i = 1)(0 - \hat{\pi}(t|X_i))^2 + I(t_i > t)(1 - \hat{\pi}(t|X_i))^2 \quad (2.21)$$

where I is an indicator of which category an individual belongs to. $I(t_i \leq t, \delta_i = 1) = 1$ means a category 1 individual that had the event occur before time t. $I(t_i > t) = 1$ means a category 2 individual that did not have the event occur before time point t.

To take censoring into account, category 1 and 2 observations are weighted by the inverse probability of censoring [44]. Category 3 observations are included in the calculation of the probability of censoring. These probabilities are Kaplan-Meier estimates of the censoring distribution, denoted as $\hat{G}(t)$. Category 1 observations are weighted by $\frac{1}{\hat{G}(t_i)}$ and category 2 by $\frac{1}{\hat{G}(t)}$. The censoring distribution $\hat{G}(t_i)$ is specific to a certain event time, t_i, of each observation, reflecting the probability of surviving just to this event time. The distribution $\hat{G}(t)$ is for all censored observations, reflecting the probability of surviving the whole follow-up period.

The Time Dependent Brier Score Under Random Censorship ($\mathrm{BS^c(t)}$), shown in Equation 2.22, combines Equation 2.21 and the inverse probability of censoring weights [44], and is given by

$$BS^c(t) = \frac{1}{N}\sum_{i=1}^{N} \frac{(0 - \hat{\pi}(t|X_i))^2}{\hat{G}(t_i)} I(t_i \leq t, \delta_i = 1) + \frac{(1 - \hat{\pi}(t|X_i))^2}{\hat{G}(t)} I(t_i > t) \quad (2.22)$$

A model making random predictions would, on average, predict $\hat{\pi}(t|X_i) = 0.5$. It follows from Equation 2.22 that $\forall i \in [1, N], \quad \hat{\pi}(t|X) = 0.5$ then $\mathrm{BS^c(t)} = 0.25$ is the $\mathrm{BS^c(t)}$ score for a randomly guessing model.

The Integrated Brier Score (IBS) extends the metric so as to provide a score for a specified time period $[t_0, t_1]$ [44]. To calculate the IBS, Equation 2.22 is integrated over the time period, as shown in Equation 2.23,

$$IBS = \int_{t_0}^{t_1} BS^c(t)\, dW(t) \quad (2.23)$$

where $W(t)$ is a weighting function used to put emphasis on different time points [44]. A common weighting function is $W(t) = \frac{t}{t_1}$, which has a linear increase of weight with time.

2.7 Explainability Methods

2.7.1 Variance of the Model Gradient

A measure for evaluating the accuracy of feature importance estimates was proposed in [45]. The method quantifies model accuracy as it degrades when features deemed important by the model are removed from the input data. Many commonly used explainability methods, such as Integrated Gradients [46] and Guided BackProp [47], were found to be worse or equal to randomly marking features as important [45]. However, the ensemble techniques SmoothGrad-Squared and VarGrad were shown to outperform random feature importance.

VarGrad is a method for interpreting CNN predictions by computing gradient-based saliency maps [48]. VarGrad works by perturbing the input and focusing on the variance in the gradients of the model's output over several perturbations. Areas of high variance are more influential on the model's predictions, and therefore provide an indicator of feature importance. VarGrad is defined in Equation 2.24.

$$\text{VarGrad} = \nu(E(x + g_i)) \tag{2.24}$$

The model explanation method, $E(x)$, where x is the model input, is perturbed by g_i, noise from a normal distribution such that $g_i \sim \mathcal{N}(0, \sigma^2)$. The standard deviation of the normal distribution σ is manually chosen when computing the VarGrad. This process is repeated a set number of times per sample. The variance, ν, is calculated from these perturbed sets of inputs, giving the VarGrad saliency map.

2.7.2 Shapley Additive Explanation

SHAP is a model interpretability method that assigns each feature an importance value for a particular prediction [49]. This importance value is based on additive feature importance measures under three properties. Additive feature importance measures, defined in Equation 2.25, express model predictions as a sum of individual feature contributions with a baseline reference value, ϕ_0, given by

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{2.25}$$

Here, $g(z')$ is the explanation model with the simplified input space $z' \in 0, 1^M$, M is the number of simplified input features. z' is a binary vector of whether a feature is included in the simplified space. The baseline reference value ϕ_0, is the value of predictions if no features were present, usually the average of all predictions. $\phi_i z'_i$ is the feature importance value, ϕ_i, of each feature i multiplied by their presence in the simplified input space z'_i. This decomposes the predictions into the contributions of the individual feature [49].

To compute SHAP values, the additive feature importance measures must fulfill three properties [49], listed below. These properties are derived from game theory and the concept of Shapley values, where the goal is to fairly distribute the "payout" (prediction) among the "players" (features) based on their contribution:

1. Local Accuracy
   For a specific input, the sum of all SHAP values, ϕ, plus the baseline reference value ϕ_0, must equal the original model prediction.

2. Missingness
   All features with no impact on prediction have no impact on SHAP values. That is, a feature that does not change the model prediction should not contribute to the SHAP explanation.

3. Consistency
   If a feature's contribution to the model prediction increases or stays the same, regardless of the other features' values, the corresponding SHAP value should not decrease.

By applying the three properties, local accuracy, missingness and consistency to the additive feature importance measure, SHAP ensures the feature importance measures are fair and accurate, reflecting the true impact of each feature on the model predictions [49].

Chapter 3

Materials and Methods

3.1 Patient Characteristics

Table 3.1: Patients characteristics for the Oslo University Hospital (OUS) and Maastro Clinic Maastricht (MAASTRO) datasets, adapted from [28].

Characteristics	OUS	MAASTRO
Number of patients		
	139	99
Age [years]		
Mean ± SD (median)	$60.2 \pm 7.7(60)$	$61.6 \pm 9.5(61)$
Gender		
Male	107 (77.0%)	73 (73.7%)
Female	32 (23.0%)	26 (26.3%)
Overall stage		
I-II	72 (51.8%)	19 (19.2%)
III-IV	67 (48.2%)	80 (80.8%)
Smoking [packs per year]		
Mean ± SD (median)	25.0 ± 22.8 (22.5)	46.1 ± 47.5 (40)
HPV-related cancer		
Yes	80 (57.6%)	22 (22.2%)
No	59 (42.4%)	77 (77.8%)
Charlson comorbidity index		
0	86 (61.9%)	25 (25.3%)
1-6	53 (38.1%)	74 (74.7%)
SUV_{peak}		
Mean ± SD (median)	11.0 ± 5.4 (10.0)	11.2 ± 6.2 (10.6)
OS		
Event	57 (41.0%)	53 (53.5%)
Non-event	82 (59.0%)	46 (46.5%)
DFS		
Event	68 (48.9%)	59 (59.6%)
Non-event	71 (51.1%)	40 (40.4%)
Proportion of censored patients		
OS	61.4%	45.6%
DFS	54.3%	40.4%

Patients with HNC from two different cohorts, a total of 238, were analyzed [28]. From Oslo University Hospital (OUS), 139 patients were collected between the years 2007 to 2013. 99 patients from the Maastro Clinic Maastricht (MAASTRO) clinic where collected between years 2008 to 2014. Patients from both cohorts who lacked contrast-enhanced CT scans, had oropharyngeal cancer, or had unknown HPV status were excluded from this thesis.[28]. Patient characteristics are shown in Table 3.1, adapted from [28].

Patients from both hospitals shared a similar age distribution, with OUS having a mean age of 60.2 with a standard deviation of 7.7, and MAASTRO having a slightly higher mean age of 61.6 with standard deviation 9.5. The gender distribution was also similar, with 77.0% being male in the OUS dataset, and 73.7% in MAASTRO.

The MAASTRO patients had a notable higher proportion of high stage cancers, where 80.8% of MAASTRO patients had cancer *Stage III-IV*, compared to the relatively even distribution in the OUS data, where 48.2% of patients have *Stage III-IV*. The MAASTRO data also had a higher number of cigarette packs smoked per year. MAASTRO patients smoked an average of 46.1 packs per year compared to 25.0 packs per year for OUS patients. This indicates that patients from the MAASTRO clinic had a higher disease severity and a larger consumption of tobacco.

On the other hand, HPV-related cancers were more common among OUS patients. 57.6% of patients from the OUS hospital had HPV-related cancer, compared to a considerably lower percentage of 22.2% in MAASTRO patients.

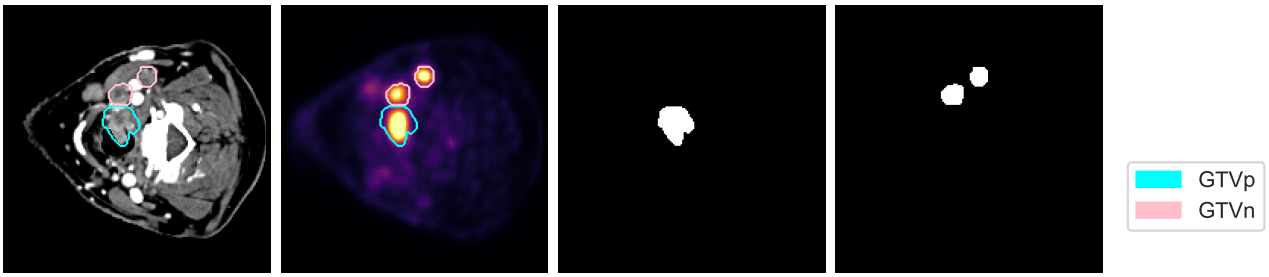
The Charlson comorbidity index is a measure of mortality risk [50]. A higher number corresponds to a higher severity of risk. The OUS dataset had a larger proportion of patients with a score of 0, 61.9%, compared to 25.3% for MAASTRO patients, where the majority had scores between 1 and 6.

The max Standardized Uptake Values (SUV), a measure of the maximum metabolic activity in the pet images [51] [10], is similar across the datasets. The OUS dataset had a mean SUV_{peak} of 11.0 with a standard deviation of 5.4, while the MAASTRO dataset had a mean SUV_{peak} of 11.2 with standard deviation 6.2.

The endpoints used in this thesis were OS and DFS. OS was defined the time from the beginning of treatment until death [28]. DFS was the time from the beginning of treatment until the first signs of recurrence of the cancer. The MAASTRO patients had a higher proportion of experienced events for both endpoints. The percentage of MAASTRO patients who experienced the OS event was 53.5% and DFS 59.6%, compared to OS 41.0% and DFS 48.9% for OUS patients.

The proportion of censored patient, that is, patients who were either lost to follow-up or had not experienced the event by the end of the study, was higher for the OUS dataset than for the MAASTRO dataset. OUS patients had a censoring percentage of 61.4% for OS and 54.3% for DFS, compared to MAASTRO's 45.6% for OS and 40.4% for DFS.

3.2 Image Modalities and Contours



(a) CT image with primary and nodal GTV contours.

(b) PET image with primary and nodal GTV contours.

(c) GTVp primary tumor mask.

(d) GTVn nodal region mask.

Figure 3.1: Image modalities, primary and nodal GTV contours. (a) shows a slice from a CT image with added primary tumor and nodal region contour. (b) shows a slice from a PET image with added primary tumor and nodal region contour. (c) shows a slice of the primary tumor contour mask. (d) shows a slice of the nodal region contour mask.

The modalities and contours used in this thesis were collected as described in [28]. CT and PET images, along with primary and nodal GTV contours, were used as inputs for the models, as shown in Figure 3.1. Images of all four types were collected from both the OUS hospital and the MAASTRO clinic.

CT images were collected following the standard procedure used for HNC radiotherapy at each hospital [28]. The PET images were 18F-Fluorodeoxyglucose (FDG)-PET scans, highlighting areas with high metabolic activity [10]. Both PET and CT images were collected using a combined PET/CT scanner. The GTV contouring was manually drawn by oncologists, in accordance with the protocols of each hospital, based on the CT and PET images [28]. GTV contours were split into one delineation of the primary tumor, called GTVp, and one nodal delineation, called GTVn.

When a GTV contour was given as input to a model, the contours were multiplied with the modalities CT and PET. This created additional masked images containing the original CT and PET information only for regions inside the delineated areas. For example, a model with the input CT, PET and GTVp will in actuality have as its input: CT images, PET images, the GTVp primary tumor contour, the CT-primary-tumor-area and the PET-primary-tumor-area.

3.3 Image Preprocessing

The modalities and contours were preprocessed before being used as input for the models. The images used in this thesis were already preprocessed for a previous study [28]. The images were co-registered so as to align the metabolic information from the PET modality with the anatomical information from CT, and to align the GTV contours. All modalities and contours were cropped to a $191 \times 265 \times 173\text{mm}^3$ volume around the tumor and nodal areas. All images were resampled to 1mm^3 isotropic voxels, meaning that the voxel represented 1mm^3 in all directions.

CT images were windowed to a center of 70 Hounsfield Units (HU) and width 200 HU, to enhance the visibility of soft-tissue [28]. This windowing approach was proven to be useful for segmentation in previous studies [52] [53]. PET images were converted from values given in Bq/mL to SUV values normalized by body weight. The PET images were cut at 25 SUV, which was the 95% percentile of max SUV values for all patients in the OUS dataset. The PET cutoff was done to eliminate outliers in SUV intensity. Before being used as model input, the voxel intensities of the CT and PET images were normalized to a range between 0 and 1.

3.4 Model Implementation

The time-to-event model architecture of this thesis was based on a treatment outcome model in [28]. The model was a CNN model, a downscaled version of EfficientNet, made to be compatible with 3D images.

3.4.1 EfficientNet

EfficientNet is a family of CNN models with a novel approach to scaling the size of the model [54]. Typically, the number of layers, called model depth, amount of neurons in a layer, called model width, and size of input layer, called resolution, are chosen arbitrarily, leading to suboptimal performances and high computational costs [54]. The depth of the network determines how complex the features the model can learn are, the width how many processes can be run simultaneously, and the resolution determines how fine the input details given to the model are.

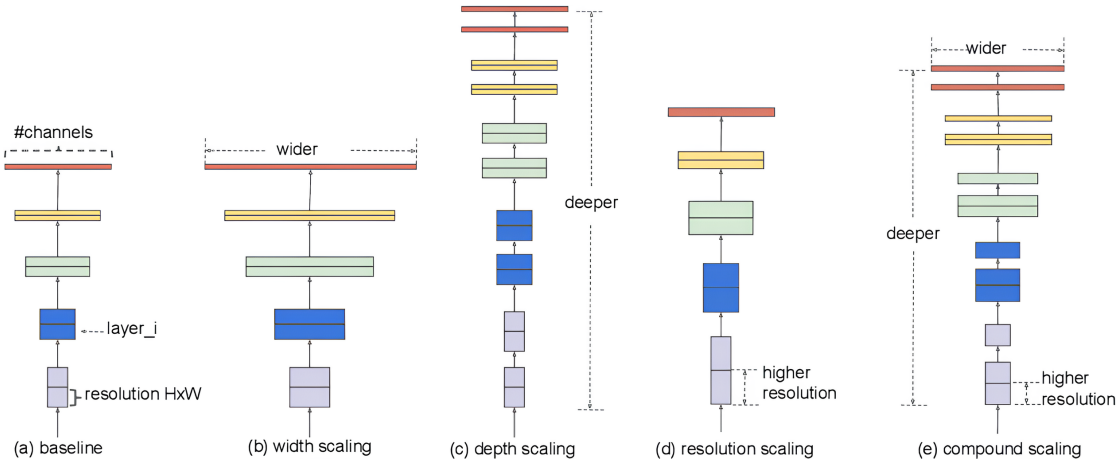


Figure 3.2: EfficientNet scaling, reproduced with permission from [54]. (a) shows an example model architecture without any scaling. (b) shows uniform width scaling, where the number of neurons in each layer is scaled up, relative to the model in (a). (c) shows uniform depth scaling, by increasing the number of layers and keeping the width and resolution the same. (d) shows resolution scaling, where only the input layer changes. (e) shows compound scaling, combining the scaling methods of (b), (c) and (d).

EfficientNet uniformly scales the depth, width and resolution [54], as seen in Figure 3.2. This is done with a user specified scaling coefficient, ϕ, that scales the size of the network, and therefore, the resources available for the model to train, as seen in Equation 3.1.

$$\begin{aligned} d &= \alpha^{\phi} \\ w &= \beta^{\phi} \\ r &= \gamma^{\phi} \end{aligned} \tag{3.1}$$

Here d is the model depth, w is the model width and r is the model resolution. The constants α, β and γ are predetermined from a small grid search and are fixed during model training [54]. Different versions of EfficientNet, ranging from the model with the fewest parameters, EfficientNetB0, to the largest model, EfficientNetB7, will have different predefined values for α, β and γ. The efficacy of EfficientNet models is largely based on the model structure determined by the constants α, β and γ, and their scaling coefficient ϕ.

3.4.2 Implementation of EfficientNet

All models were implemented through the Deoxys framework, available at `https://pypi.org/project/deoxys/`. Deoxys is a framework designed to streamline the application of deep learning techniques, with emphasis on medical images and cancer segmentation. The code for the model implementation and analysis can be found at `https://github.com/huynhngoc/hnc-surv`. Important code excerpts are found in Appendix A.

This thesis used a custom downscaled 3D version of EfficientNetB1 for all models. A 3D version of EfficientNetB1 is severely limited by computational costs because of the higher dimensionality of 3D images [55], and therefore needed to be downscaled to reduce the cost. The EfficientNet architecture was chosen because it had proven to outperform other CNN models while having fewer model parameters [54], reducing the computational cost of the model. The downscaling involved reducing the number of filters in each convolutional layer by half, setting the scaling coefficient ϕ to 0.5 for the model width. The 3D implementation was done by replacing all 2D layers with 3D layers from the 2D EfficientNet found in the TensorFlow library version 2.11 [56].

The model in [28] was designed for treatment outcome predictions and was not capable of performing time-to-event predictions. The model was made compatible with survival time estimation by changing the loss function to a negative log likelihood loss function. This new loss function was compatible with time-to-event analysis by quantifying the loss based on survival status in discrete time intervals, and incorporating censoring information. To make survival predictions over time, the final layer of the model was changed into a dense layer. This layer had the number of neurons equal to the number of time intervals. This made the model output a vector with size equal to the number of time intervals, with each item giving a probability of surviving through that interval.

The models were initialized with the Adam optimizer from the Keras library version 2.11 [57], with a learning rate of 0.0001. Adam is an efficient optimizer that adapts the learning rate by using the average and variance of the gradients of the loss function, and adapts a new learning rate to each weight in the model [33].

3.4.3 Data Augmentation

Table 3.2: Data augmentation techniques and parameters, with their respective probability of being applied.

Augmentation Technique	Value	Probability
Rotation	Between -15° and 15°	20%
Rescaling	Factor from 0.9 to 1.1	20%
Shifting	Range of 5 voxels in each direction on each axis	10%
Flipping	Inverting the image in the sagittal plane	50%

The input images were augmented before model training, following the method in [28]. The data augmentation techniques are summarized in Table 3.2. Data augmentation involves applying various transformations to the input to mitigate overfitting and generalize the model [58]. Data augmentation was performed using the *ClassImageAugmentation3D* function from Deoxys. The input images, being 3D volumes, were rotated with a range of $\pm 15°$ around their three axes. The images were rescaled with a factor between 0.9 and 1.1, and shifted with a range of 5 voxels in each direction on each axis. The images were also flipped in the sagittal plane. The probability of an image being augmented during training was 20% for the rotation, 20% for the rescaling, 10% for the shifting and 50% for the flipping.

3.4.4 Train/Test Scheme

The general workflow for training and testing the models was adapted from [28]. A total of 13 models were developed, each representing a different combination of modalities, contours, endpoints and time intervals, summarized in Table 3.3.

Table 3.3: Model input combination scheme. Various combinations of modalities, contours, endpoints and time intervals were used to make 13 models. Eight models were made for the OS endpoint and five models were made for the DFS endpoint.

Model Input	Time Intervals
Overall Survival	
CT	10
PET	10
PET	20
CT+PET	10
PET+GTVp	10
CT+PET+GTVp	10
PET+GTVp+GTVn	10
CT+PET+GTVp+GTVn	10
Disease Free Survival	
CT	10
PET	10
CT+PET	10
PET+GTVp	10
CT+PET+GTVp	10

All models, except one, were evaluated using 10 time intervals. One PET-only model was tested using 20 time intervals to assess the impact of splitting the follow-up period into a larger number of smaller intervals.

The combinations of modalities for each model was chosen to cover a broad spectrum of information combinations. CT and PET images were combined in various ways with GTV contours to leverage the anatomical and metabolic information they provide. The hypothesis was that the spatial resolution of CT, the metabolic information from PET and the exact delineations from GTV contours could be combined to offer a more nuanced understanding of tumor behavior. Single and dual modality models were also made to assess the performance of simple models that were not given as much information about the cancer.

The primary endpoint for the evaluation of these models was OS. The four models with the highest predictive performance and one model with the lowest performance, as measured by the C-index, were used for predicting the DFS endpoint. The DFS endpoint was used to further asses models of interest since this endpoint can be harder to predict than OS [28] [51].

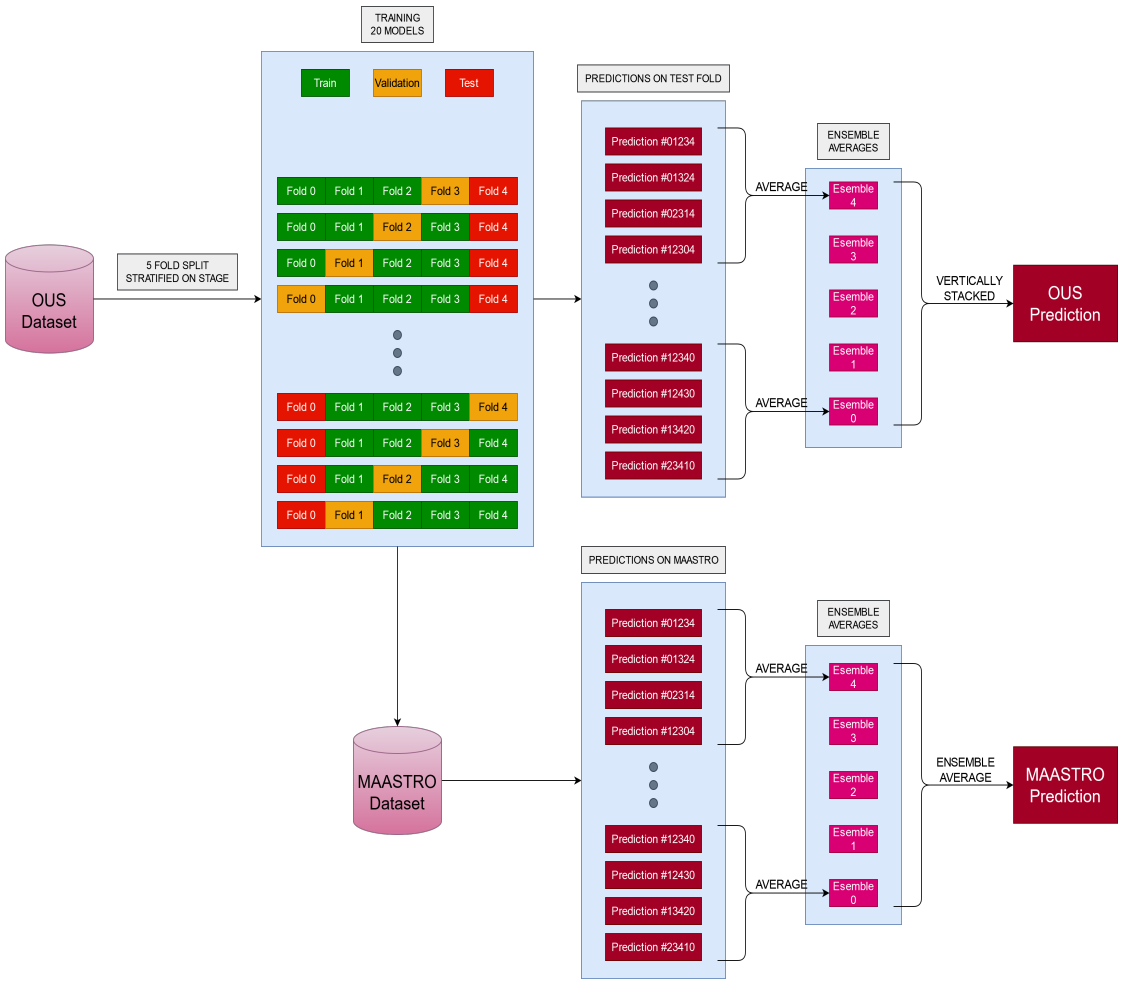


Figure 3.3: Model training and testing scheme, based on [28]. The cross-validation scheme involved splitting the OUS data into train, test and validation folds. Training a model for each fold combination, with total of 20 combinations. Combining each model with the same test fold into ensemble models and averaging. Finally vertically stacking the ensemble models to obtain the final OUS prediction scores. The same fold combination models were trained on the external MAASTRO dataset. Each model was averaged into ensemble models and averaged again to obtain the final MAASTRO prediction scores.

All models went through the same nested five-fold cross-validation scheme, adapted from [28], as seen in Figure 3.3. This was implemented so that the model performances were robust, and that they generalized well to unseen data [59]. Data from the OUS dataset was used for training and validating the model, and the MAASTRO data was used as an external test set, unseen by the model during training.

First, the OUS training data was split into five folds, as seen at the *5 Fold Split* part of Figure 3.3. These folds were stratified on cancer stage, keeping the proportion of *Stage I-II* and *Stage III-IV* the same in each fold as in the whole dataset. The proportion of stage was kept, following the scheme in [28], because it was found to be an important factor for outcome prediction. Each fold then contained 27 or 28 patients from the 139 total. Each model used three folds for training, one for validation during training, and one for testing. This resulted in 20 unique combinations of folds, as seen in the *Training 20 Models* part of Figure 3.3. One model was trained and tested per fold combination, resulting in 20 models. These 20 models all had the same input combination of modalities and contours.

Each model was run for a total of 60 epochs, calculating performance with the validation fold every epoch after the 20^{th}. To not exceed memory restrictions, the batch size was set to 4, meaning that four patients were used from the training folds per training iteration. From the 20^{th} epoch, model weights and performance metrics were saved. The metrics calculated were C-index, AUC and IBS. The model weights corresponding to the epoch with the highest C-index were selected as the model representing that fold combination. These predictions are seen in the *Predictions on Test Fold* part of Figure 3.3.

All models using the same test fold were averaged into one ensemble model, seen in the *Ensemble Averages* part of Figure 3.3. This resulted in five ensemble model averages, with each giving one prediction per patient in their respective test fold. These ensemble averages were stacked vertically, resulting in one set of prediction metrics for all patients in the OUS dataset, called *OUS Prediction* in Figure 3.3.

The same 20 models were used to predict the external MAASTRO dataset, as seen coming off the *Training 20 Models* part of Figure 3.3, providing a measure of the model's stability and generalizability. Each model predicted on all MAASTRO patients, which resulted in 20 sets of full predictions, seen in the *Predictions on MAASTRO* part of Figure 3.3. The prediction sets were averaged into five ensemble models, and then averaged again to get one set of predictions for the MAASTRO dataset, shown in the *Ensemble Averages* and *MAASTRO Prediction* parts of Figure 3.3.

All models were run on the Orion High Performance Computing cluster at the Norwegian University of Life Sciences. ORION can be found at `https://orion.nmbu.no` (internal). The ORION GPU capacity consists of four nodes, each with three Nvidia Quadro RTX 8000 – 48GB GPUs. A model for one fold combination took around six to eight hours to run. Four fold combination models were run in parallel, making the total time to train one full model 30 to 40 hours.

3.5 Implementation of the Negative Log Likelihood Loss Function

The implementation of the log likelihood loss function $\log L$ is based on [16], and is defined in Equation 3.2. The code excerpt of the negative log likelihood loss function can be found in Appendix A.1.

$$\log L = -\sum_{i=1}^{N} \ln(1 + s_i \cdot (o_i - 1)) + \ln((1 - f_i) \cdot o_i) \tag{3.2}$$

Here, N is the total number of individuals, with i being one individual. The output of the model is stored in the vector o, and contains the probabilities of the individual not having the event occur in each time interval, that is, surviving through the interval. Any item in o ranges from 0 to 1, where 0 is 0% predicted chance of surviving through the interval, and 1 is 100% predicted chance of surviving. The vector s indicates the time intervals in which the individual i has had no event occur. Vector s has value 1 for intervals where the event did not occur, and value 0 in the interval the event occurred and afterwards. The vector f indicates where the event occurred and has a value of 1 in the time interval where the event occurred, and 0 in all other intervals.

The vectors s and f were made from the OUS and MAASTRO datasets, using the *MakeSurvArray* function, found in Appendix A.2. First, the true time and true event information was extracted from the data. The number of samples and the number of intervals were calculated, based on the predefined split of the follow-up period, given as interval break points. Then, the size and midpoint of each time interval was computed. Using this, the vectors s and f were made. If a sample had the event occur, the vector s was set to 1 for intervals that passed with no event, that is, where the time until event was greater or equal to the upper limit of the given interval break. The vector f was set to 1 where the time until event was less than an upper interval break. If a patient was censored and had no event, the s vector was set to 1 for intervals where the time until censoring was greater than or equal to the midpoint of each interval. This means that censored patients were given credit for surviving through an interval if the time until censoring was greater than halfway through the interval. Every entry in the f vector was 0 for censored patients.

The model concatenated the s and f vectors into one *SurvArray* tensor, and appended to the end the true event and time information used for metric calculation. For example, a division of ten time intervals means a tensor of shape $N * 22$, since the vectors s and f have the length of the number of intervals, and the time and event information have length 2, where N is the number of samples.

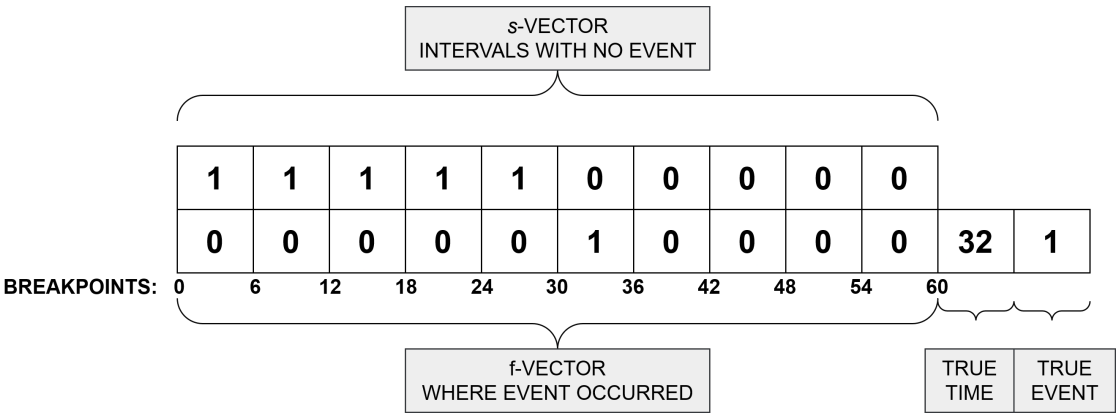


Figure 3.4: Example *SurvArray* tensor for one sample, ($N = 1$), with ten time intervals. The breakpoints making up the ten time intervals are shown under the tensor, from 0 months to 60 months. The patient experienced the event within the first half of the 6^{th} time interval, and was not given credit for surviving through the interval. The first ten elements constitute the s vector, and show the patient as having survived through the first five time intervals. The next ten elements make up the f vector, which shows the event occurring in time interval 6. The last two elements contain the actual survival time and event information, which show a survival time of 32 months and that the event occurred.

Figure 3.4 shows an example tensor for one sample with ten time intervals for an individual who had the event occur within the 6^{th} interval. The s vector indicates that no event occurred in the first five time intervals, and the f vector indicates the event occurred in the 6^{th} interval. True time and event information is included in the last two elements of the tensor.

For all but one of the models, the follow-up period was split into ten time intervals. The break points for the time intervals, in months, were set to:
0, 6, 12, 18, 24, 30, 36, 42, 48, 54, 60. This makes the first time interval between 0 and 6 months, the second time interval between 6 and 12 months, and so on. These intervals were spaced with six months, focusing on the beginning of the follow-up period.

To assess different interval sizes and number of intervals, a model was trained on 20 intervals with the break points:
0, 6, 8, 12, 16, 20, 24, 32, 39, 42, 45, 49, 55, 59, 64, 67, 69, 76, 81, 85, 90. The 20 intervals were spaced out to include around the same number of patients in each interval and to be more equally spaced over the follow-up period. This was done to avoid bias on any interval or set of intervals.

3.6 Implementation of the Evaluation Metrics

3.6.1 AUC

The AUC metric was calculated using the *roc_auc_score* from the Scikit-learn library version 1.3.0 [60]. The code excerpt for AUC calculation is found in Appendix A.3. The function required two inputs: an array of true labels and an array of corresponding estimates.

The array of true labels was found in the s vector, which gave a 1 if an individual has survived an interval, 0 if not. The model extracted the s vector from the *SurvArray* tensor, taking the n first items, where n is the number of time intervals. The model predictions was found in the o vector, which recorded the model output, that is, the predicted probability for surviving each intervals.

3.6.2 Harrel's Concordance Index

The C-index was computed using the function *concordance_index* from the lifelines library version 0.28.0 [61]. The code excerpt for C-index calculation is found in Appendix A.4. The function required true time until event, predicted score, and true event information. The true time until event and true event information was extracted from the last two items in the *SurvArray* tensor.

The predicted score was calculated from the vector o, by taking the cumulative product of predicted probabilities up to a specified period. This resulted in a single risk score for each individual, representing the predicted survival probability at the end of the specified period. The specified period was set to five years, making the C-index a measure within a specific time frame. The five year period corresponded to the ten time intervals, where 60 months (five years) was the last breakpoint. This method assumed that the cumulative product of probabilities up to a certain interval represents the survival probability up to that time.

3.6.3 IBS

IBS was calculated using the *integrated_brier_score* function from the Scikit-survival library version 0.22.2 [62]. The code excerpt for IBS calculation is found in Appendix A.5. The function required four parameters: (1) an array of true survival times and event information of the training data, used to estimate the censoring distribution; (2) an array of true survival times and event information of the test data; (3) an array of estimated survival probabilities at specified time points; (4) specified time points to estimate the BS at.

The model extracted the true time and event information from the last two items in the *SurvArray* tensor and listed them together. This list served as both the array used to estimate the censoring distribution (1), and the array of true survival times and event information (2). The estimated survival times (3) were the model outputs found in vector o. The specified time points (4) were the provided break points. Since the provided break points stop at 60 months (IBS was not calculated for the 20 interval model) the metric quantified prediction error over a five year period.

3.7 Implementation of KM Curves

KM curves were made using the *KaplanMeierFitter* function from the lifelines library version 0.28.0 [61]. To draw the curves, two sets of data were needed: the observed true time-to-event data with clinical information, and the time-to-event model predictions. Clinical data, like cancer stage or HPV status, was merged with the actual and predicted time-to-event data. This new dataset was split into groups based on some clinical factor, for example, one group with cancer *Stage I-II*, and one with cancer *Stage III-IV*.

The *KaplanMeierFitter* was now initialized, one for each group. The curve for the observed true data, called the ground truth, was made by fitting the KM estimator to the observed time-to-event data, and observed event information. The curve was only drawn at time intervals that corresponded to the chosen break points. The curve was drawn using the *plot_survival_function* in the *KaplanMeierFitter*. This function included an upper and lower confidence interval, showing the uncertainty around the survival estimate. The confidence interval was calculated by Greenwood's Exponential formula [61], which approximates a 95% confidence interval [63]. Risk counts, a count of individuals at risk of experiencing the event, censored individuals and individuals who have experienced the event, at each interval, was added to the plot. The risk counts were made by the *add_at_risk_counts* function.

Estimated KM curves were made by initializing a new set of *KaplanMeierFitter*s. These were fitted to the predicted time-to-event and predicted event status. The curves were drawn using the same functions as for the ground truth curves, with added confidence intervals and risk counts.

A log-rank test was made after the curves were estimated. The log-rank test was calculated with the *pairwise_logrank_test* function from the lifelines library version 0.28.0 [61]. The test was given the same time until event, event status and groups as the *KaplanMeierFitter*. This gave log-rank tests for the observed and predicted KM curves.

3.8 Implementation of Explainability Methods

The VarGrad and SHAP methods were chosen for assessing how different features of the input data contributed to the survival predictions. Gradient based explainability methods are often noisy [64], and can highlight seemingly arbitrary voxels. VarGrad has proven to be superior to other gradient based methods [45], and can make the saliency hap less noisy by averaging over a chosen number of repetitions per image. The SHAP method was chosen for its game theory approach, ensuring fair and additive explanations [49].

3.8.1 The VarGrad Method

VarGrad heatmaps were made by calculating the variance of the model gradients when the image was perturbed by noise. The noise was normally distributed with a standard deviation of 0.05. The calculation was repeated a number of times for each image, making the heatmap smoother and less noisy [48]. VarGrad was calculated with 20 repetitions per image.

A series of statistical plots were made to assess the relation between the VarGrad heatmaps and the input data. The plots compared the OUS and MAASTRO datasets to the VarGrad heatmaps, assessing the variability and consistency in model predictions.

Two plots were line graphs displaying the relationship between mean VarGrad values and HU values from CT images, as well as between mean VarGrad values and SUV values from PET images. First, the mean VarGrad values for each SUV and HU value were calculated. These mean VarGrad values were then plotted against the corresponding SUV and HU values. The resulting graphs were used to evaluate what metabolic activity and anatomical data had predictive influence.

A histogram showing the VarGrad values found within certain areas was made. This plot showed the mean VarGrad values found within the primary tumor, nodal areas and outside any delineated area. This plot was used to assess in which of the areas, delineated by the GTVp and GTVn contours, the VarGrad heatmap was present in, and therefore, which areas influenced the model predictions.

The final plot was a violin plot showing the correlation of VarGrad values and SUV values for all patients within the datasets. Violin plots show the distribution of correlation coefficients with an inner box plot. This plot highlights the variability of the correlation of VarGrad values with SUV values across the patients from both datasets.

3.8.2 The SHAP Method

Several types of SHAP explainers can be utilized on images, but the limitation of requiring the model input to be 3D volumes of shape $number\ of\ samples \times depth \times height \times width \times channels$ limited the number of choices. Gradiant SHAP, from the SHAP library version 0.44.11 [49], was the only explainer found to be compatible with the models used in this thesis, and then, only the single-modality models, where the shape the number of channels was 1 due to there only being one modality as input.

Gradient SHAP is a method based on Integrated Gradients [46] and is therefore similar to VarGrad in that it is based on the model gradients. Gradient SHAP calculates the expected values of gradients by sampling from a baseline distribution, similar to Integrated Gradients. First, a baseline of samples is chosen as a background distribution. Then, input data is perturbed relative to the baseline samples, meaning features from the input are gradually introduced to the baseline. For each perturbed sample the gradient is calculated and scaled to fulfill the SHAP properties, attributing the output prediction to each input feature based on its contribution.

One fold from the dataset was used in calculation of the SHAP values. This one fold array was reduced to a one-channel PET-array containing only the PET data, corresponding to the PET-only model, and preprocessed according to the modality. From the PET-array, a 4D array was extracted for a specified patient. Since there was only one channel, the PET channel, this patient array could be represented as a 3D volume of shape $depth \times height \times width$, compatible with the Gradient SHAP explainer.

A *GradientExplainer* was initialized from the SHAP library version 0.44.1 [49]. The explainer was given the PET-only model and the PET-array for all patients in the fold, used as the background dataset.

The SHAP values were calculated for the patient array. The images for the patient were perturbed 100 times in the calculation. One SHAP heatmap was created per model output node, which corresponded to ten heatmaps for the ten interval model.

Thresholded SHAP values were created by binarizing the SHAP values against a threshold. The threshold was set at the 99^{th} percentile of significance, meaning the top 1% significant SHAP values were set to 1, the others to 0. This resulted in a binary mask of the top 1% significant SHAP values.

The built in plotting function *shap.image_plot* was used to create the SHAP plots. Both the raw SHAP values and the thresholded SHAP values were plotted, for each time interval. The plotting function requires both the original image and SHAP values to be a 2D image with three channels, like that of an RGB image. Since different SHAP values were made for each model output, here, time intervals, one image was made per interval. For each interval a slice from the 3D volume of the patient and of the corresponding SHAP values were extracted. The slice and values were both converted into pseudo-RGB by repeating them three times, creating three channels. Then the pseudo-RGB image slice and pseudo-RGB SHAP values were given to the plotting function, and the GTV contours were overlaid to provide information of the overlap of SHAP values in the tumor and nodal areas.

3.9 AI statement

The use of AI in this thesis followed the current regulations as of May 2024 at the Faculty of Science and Technology (REALTEK), found at https://www.nmbu.no/fakulteter/fakultet-realfag-og-teknologi/kunstig-intelligens-ved-realtek.

ChatGPT, a language model by OpenAI, found at https://openai.com/chatgpt, was used in the creation of this thesis.

ChatGPT was used to format LaTeXtables and aid in formatting. Example prompts include: "Make the cell text span all cells in the table" and "Add vertical and horizontal lines, in gray, for the white cells".

ChatGPT was used in code development. It was used to write small code snippets and to debug existing code. Example prompts include providing the code with the error message.

ChatGPT was used to summarize papers to quickly get an overview of their contents. It was not used as the source for information, rather to find out which papers were worth pursuing in depth. Care was taken to fact check every claim made by the ChatGPT model. Example prompts include giving the ChatGPT model one or more papers in the form of PDF files, and asking it to summarize them.

ChatGPT was used to give suggestions on how to rewrite portions of text for clarity. This was done while making sure the ChatGPT model did not add nor subtract any information. Example prompts include giving the ChatGPT model a paragraph of text and asking it to rewrite the text while not adding or subtracting information. Care was taken to ensure the rewrites were factual and correct.

Chapter 4

Results

4.1 Model Performances

4.1.1 Predictions on the OUS Dataset

Table 4.1: Model predictions on the OUS dataset, sorted by C-index. The highest C-index performing models are highlighted in bold. All models were run on 10 time intervals.

Model	C-index	AUC	IBS
Overall Survival			
CT+PET+GTVp	**0.74**	**0.69**	**0.16**
PET+GTVp	0.72	0.68	0.16
PET+GTVp+GTVn	0.71	0.67	0.16
PET	0.70	0.63	0.17
CT+PET	0.66	0.61	0.18
CT+PET+GTVp+GTVn	0.66	0.62	0.17
CT	0.61	0.56	0.18
Disease Free Survival			
PET	**0.62**	**0.55**	**0.23**
PET+GTVp	0.60	0.54	0.24
CT+PET+GTVp	0.59	0.54	0.23
CT+PET	0.59	0.54	0.23
CT	0.51	0.49	0.24

Table 4.1 shows the performances of the various ensemble model averages tested on the OUS dataset. The ensemble model results are found in Appendix B.1. The models are sorted by their C-index with the highest performing models in bold. The models were evaluated on three metrics: C-index, AUC and IBS, and on two endpoints: OS and DFS.

For the OS endpoint, the CT+PET+GTVp model had the highest C-index, highest AUC score and the lowest IBS error, making it the highest performing model on all metrics. Models including the CT modality generally performed lower on all metrics, with the exception of the highest performing model. In addition, a PET-only model on the OS endpoint, not included in the table, was trained using 20 time intervals and achieved a C-index of 0.69, which was similar to the 10 time interval PET-only model shown in Table 4.1, which achieved a C-index of 0.70.

For the DFS endpoint, the four OS models, with the highest C-index, were examined, with the inclusion of a CT-only model. The PET-only model performed the best on all metrics with a C-index of 0.62, AUC of 0.55 and IBS of 0.23. Models on the DFS endpoint generally had a lower performance than models on the OS endpoint, with lower C-index and AUC scores, and higher IBS error. Similar to the OS endpoint, models with the CT modality performed worse on the C-index than models without. The CT-only model had a C-index of 0.51, AUC of 0.49 and IBS of 0.24. This is close to the expected score of a randomly guessing model, which would be expected to have a C-index of 0.50, AUC of 0.50 and IBS of 0.25 [41] [38] [44].

4.1.2 Predictions on the MAASTRO Dataset

Table 4.2: Model predictions on the external MAASTRO dataset, ordered as in Table 4.1. The highest C-index performing models are highlighted in bold. The increase or decrease in C-index from the OUS predictions is shown as *Difference in C-index*. All models were run on 10 time intervals.

Model	C-index	AUC	IBS	Difference in C-index
Overall Survival				
CT+PET+GTVp	**0.68**	**0.68**	**0.17**	**−0.06**
PET+GTVp	0.65	0.65	0.17	−0.07
PET+GTVp+GTVn	0.66	0.64	0.17	−0.05
PET	0.67	0.64	0.17	−0.03
CT+PET	0.63	0.62	0.17	−0.03
CT+PET+GTVp+GTVn	**0.69**	**0.67**	**0.16**	**+0.03**
CT	0.62	0.60	0.18	+0.01
Disease Free Survival				
PET	**0.67**	**0.63**	**0.21**	**+0.05**
PET+GTVp	0.61	0.63	0.22	+0.01
CT+PET+GTVp	0.63	0.64	0.21	+0.04
CT+PET	0.65	0.65	0.21	+0.06
CT	0.64	0.64	0.21	+0.13

Table 4.2 summarizes the ensemble model average performances, tested on the external MAASTRO dataset. The ensemble model performances are found in Appendix B.2. The models with the highest C-index are highlighted in bold. In addition to the metrics, the difference in C-index between the MAASTRO and the OUS predictions is shown, i.e. $\text{C-index}_{\text{MAASTRO}} - \text{C-index}_{\text{OUS}}$, and shows the increase or decrease in C-index when going from predicting on the OUS dataset to the external MAASTRO dataset.

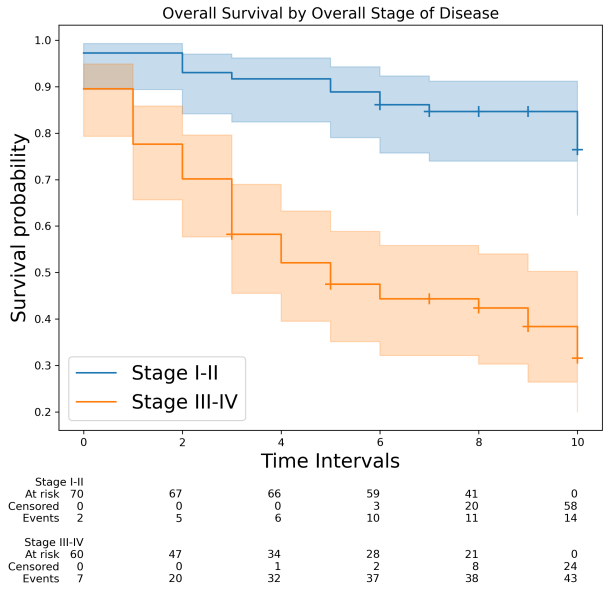
For predictions on the OS endpoint, the CT+PET+GTVp+GTVn model had the highest C-index at 0.69, an increase of 0.03 from the OUS performance in Table 4.1. Comparably, the CT+PET+GTVp, the best model on the OUS dataset, had a C-index of 0.68. The CT+PET+GTVp model had a C-index decrease of 0.06 relative to the OUS performance. There was no longer a clear separation in C-index scores for models with the CT modality for the models tested on the MAASTRO dataset, as was seen in Table 4.1 when predicting on the OUS dataset.

The PET-only model had the highest C-index on the DFS endpoint with a C-index of 0.67, giving a 0.05 increase from the OUS performance. All models on the DFS endpoint showed an increase in the C-index score when predicting on the MAASTRO dataset. The model with the largest increase was the CT-only model, with a C-index of 0.64, gaining 0.13 from the OUS C-index. Again, there was no clear separation in performance for models with or without the CT modality.

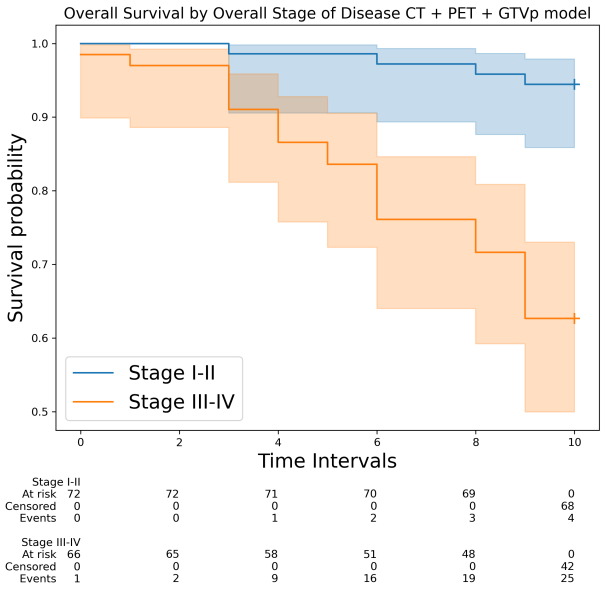
Model performances were more similar between endpoints when predicting on the MAASTRO data than on the OUS data. The C-index performances of the models tested on the MAASTRO dataset generally decreased when predicting the OS endpoint, and increased on the DFS endpoint relative to OUS performances.

4.2 Kaplan Meier Curves

4.2.1 Overall Stage of Disease



(a) Observed ground truth for the OUS data.



(b) Model prediction for the OUS data.

Figure 4.1: KM curves grouped by overall stage of disease for the OS endpoint using the CT+PET+GTVp model. The KM curves show the survival probability of two groups, the blue curve with *Stage I-II* cancer, and the orange curve with *Stage III-IV*. The figure is divided into two panels, (a) shows the observed survival present in the OUS dataset, called the ground truth, (b) shows the model estimated survival for the OUS dataset. The shaded area around the curves represent the 95% confidence intervals, which indicates the precision of the survival estimate, the narrower the confidence interval, the more precise the estimate is. Note that the panels have different y-axes, (a) ranging from 0.20 to 1.0, and (b) ranging from 0.50 to 1.0. Under the KM curves in panel (a) and panel (b) are shown the number of patients at risk, and the events and censoring counts for each time interval. For example, at time interval 1 in Figure 4.1a, 70 patients are at risk, and two have already had the event occur in group *Stage I-II*, while 60 patients are at risk, with seven having had the event occur in the *Stage III-IV* group.

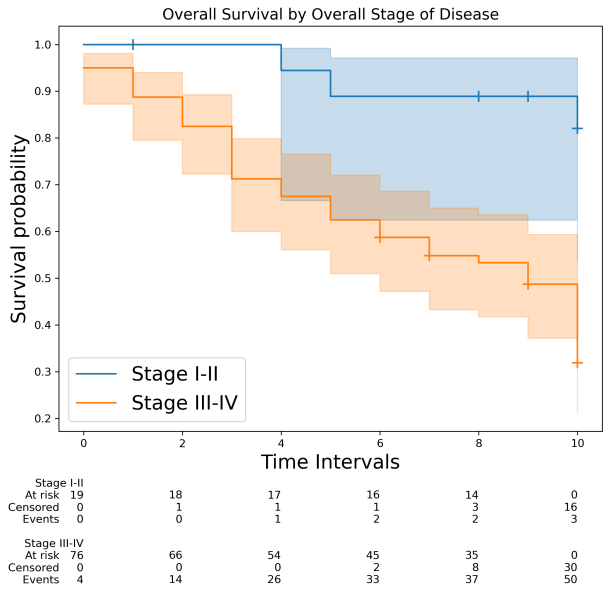
Table 4.3: Log-rank test for the KM curves given in Figure 4.1.

Ground Truth OUS			**Predicted OUS**		
Test statistic	p-value	$-log_2(p)$	Test statistic	p-value	$-log_2(p)$
31.14	< 0.005	25.31	21.46	< 0.005	18.08

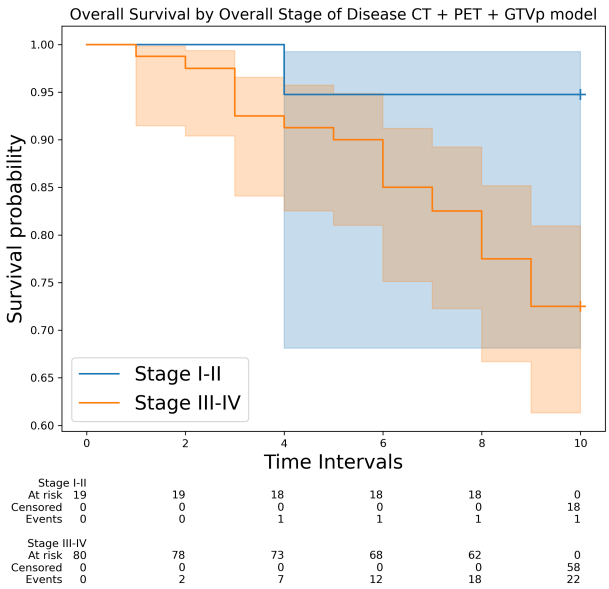
Figure 4.1 shows KM curves grouped by overall stage of disease on the OS endpoint. The survival curves span the ten time intervals the model used for prediction, which is over a five year period, where time interval 10 equals 60 months. The y-axes are different between the two panels Figure 4.1a and Figure 4.1b. Figure 4.1a ranges from 0.20 to 1.0, and Figure 4.1b ranges from 0.50 to 1.0. The KM curves in Figure 4.1a had the observed survival probabilities around 0.80 and 0.30 for *Stage I-II* and *Stage III-IV* patients respectively at the end of time interval 10. On the other hand, the model estimated survival, in Figure 4.1b, had an estimated probability around 0.95 and 0.60 for *Stage I-II* and *Stage III-IV* patients respectively at the last time interval. This shows that the model estimated higher survival than the observed ground truth. The number of patients that experienced the event at time interval 10 were, for the observed ground truth, 14 patients with *Stage I-II*, and 43 patients with *Stage III-IV*. The model estimates were 4 events for patients with *Stage I-II*, and 25 events for patients with *Stage III-IV*, at time interval 10.

Figure 4.1b shows the model predictions for the patients in each stage group. The model has been given no cancer stage information except for what it extracted from the input images. The predictions followed a similar patters to the ground truth in Figure 4.1a, where *Stage I-II* patients had a higher survival probability than *Stage III-IV*. The ground truth showed a higher distinction between the groups than the model predictions. Until around time interval 4, the model predictions did not show much separation of each group's survival probability.

Table 4.3 shows the log-rank test results corresponding to the KM curves in Figure 4.1. For the log-rank test, the significant threshold for the $p - value$ was set at 0.05. The ground truth showed a significant difference between the cancer stage groups with a $p - value$ under the 0.05 threshold. Similarly, the model predictions had a significant difference between the groups, with a similar $p - value$, showing the model's ability to differentiate between the survival probabilities of the two cancer stages.



(a) Observed ground truth for the MAASTRO data.



(b) Model prediction for the MAASTRO data.

Figure 4.2: KM curves grouped by overall stage of disease for the OS endpoint using the CT+PET+GTVp model. The KM curves show the survival probability of two groups, the blue curve with *Stage I-II* cancer, and the orange curve with *Stage III-IV*. The figure is divided into two panels, (a) shows the observed survival present in the MAASTRO dataset, called the ground truth, (b) shows the model estimated survival for the MAASTRO dataset. The shaded area around the curves represent the 95% confidence intervals, which indicates the precision of the survival estimate, the narrower the confidence interval, the more precise the estimate is. Note that the panels have different y-axes, (a) ranging from 0.20 to 1.0, and (b) ranging from 0.60 to 1.0. Under the KM curves in panel (a) and panel (b) are shown the number of patients at risk, and the events and censoring counts for each time interval.

Table 4.4: Log-rank test for the KM curves given in Figure 4.2.

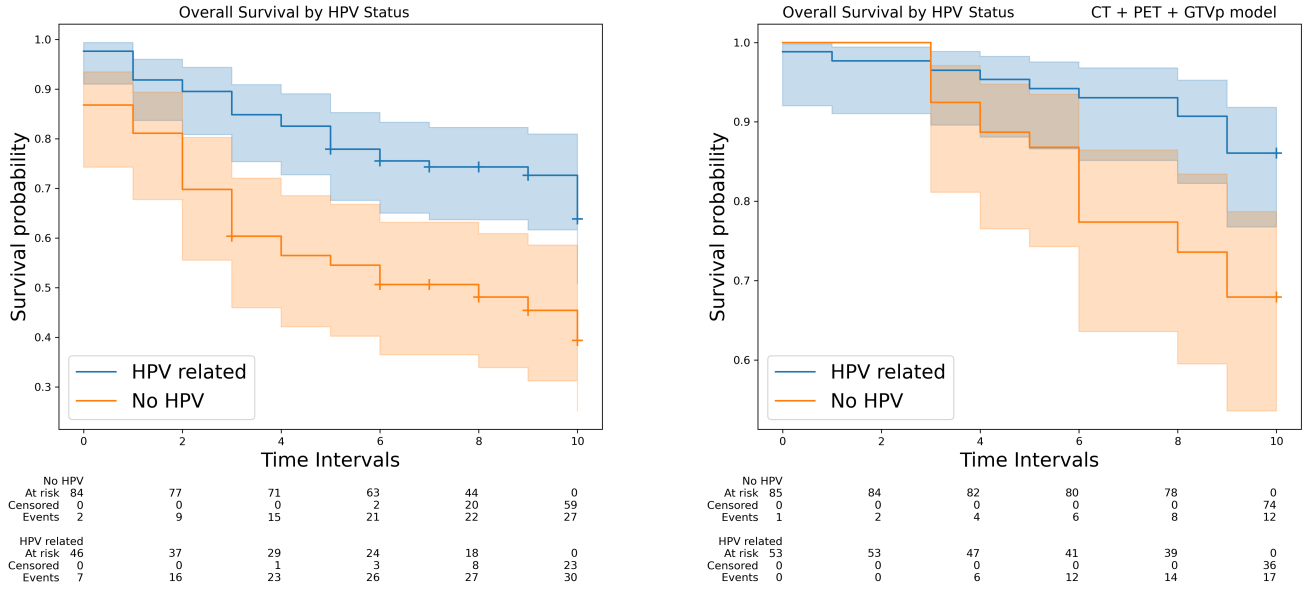
Ground Truth MAASTRO			**Predicted MAASTRO**		
Test statistic	p-value	$-log_2(p)$	Test statistic	p-value	$-log_2(p)$
11.33	< 0.005	10.35	3.84	0.05	4.32

Figure 4.2 shows KM curves for the external MAASTRO dataset, grouped by overall stage of disease on the OS endpoint. The figure shows the observed ground truth of the dataset, and the predicted survival probabilities of the CT+PET+GTVp model, in Figure 4.2a and Figure 4.2b, respectively. The KM curves span 10 time intervals used by the model for prediction across a five-year period, with the final interval corresponding to 60 months. The y-axes in Figure 4.2a and Figure 4.2b differ in scale. Figure 4.2a ranges from 0.20 to 1.0, and Figure 4.2b ranges from 0.60 to 1.0. The curves showed observed survival probabilities of approximately 0.80 for *Stage I-II* patients and 0.30 for *Stage III-IV* patients at the end of the 60 months. Figure 4.2b, the model-estimated survival probabilities, showed about 0.95 for *Stage I-II* patients and 0.75 for *Stage III-IV* patients at the same interval, indicating that the model predicted higher survival rates than the observed ground truth. The number of patients who experienced the event by time interval 10, in the observed ground truth, were 3 for *Stage I-II* and 50 for *Stage III-IV*, while model estimated 1 and 22 respectively.

Figure 4.2a shows the observed ground truth data, where the survival probability was higher for *Stage I-II* patients than *Stage III-IV*. This trend continued throughout the time intervals. From around time interval 4 to 9, the confidence intervals overlapped, meaning that it was possible that the curves were not separate during this time. Still, the $p - value$, seen in Table 4.4, was under 0.005, and the curves were therefore significantly different.

In Figure 4.2b, showing the model predictions on the MAASTRO dataset, the confidence interval of the *Stage I-II* group overlapped with the *Stage III-IV* curve. This is reflected in the $p - value$ in Table 4.4, which was 0.05. Although this $p - value$ is at the set threshold, and therefore signified a marginal significance, it suggests that the model is not capable of distinguishing between the stage groups with a high degree of confidence. Again, both model predicted curves showed a higher survival probability than the observed ground truth.

4.2.2 HPV-Positive Oropharyngeal Tumors



(a) Observed ground truth for the OUS data.

(b) Model prediction for the OUS data.

Figure 4.3: KM curves grouped by HPV status for the OS endpoint using the CT+PET+GTVp model. The KM curves show the survival probability of two groups, the blue curve with *HPV related* tumors, and the orange curve with *No HPV* relation. The figure is divided into two panels, (a) shows the observed survival present in the OUS dataset, called the ground truth, (b) shows the model estimated survival for the OUS dataset. The shaded area around the curves represent the 95% confidence intervals, which indicates the precision of the survival estimate, the narrower the confidence interval, the more precise the estimate is. Note that the panels have different y-axes, (a) ranging from 0.30 to 1.0, and (b) ranging from 0.50 to 1.0. Under the KM curves in panel (a) and panel (b) are shown the number of patients at risk, and the events and censoring counts for each time interval.

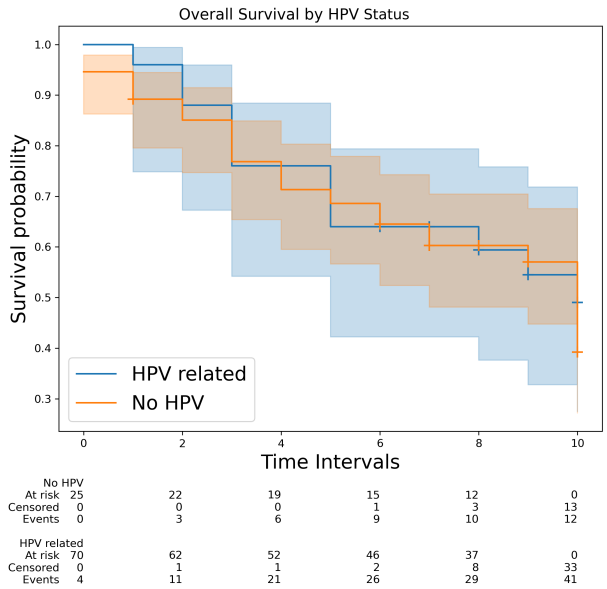
Table 4.5: Log-rank test for the KM curves given in Figure 4.3.

Ground Truth OUS			**Predicted OUS**		
Test statistic	p-value	$-log_2(p)$	Test statistic	p-value	$-log_2(p)$
10.48	< 0.005	9.69	6.74	0.01	6.73

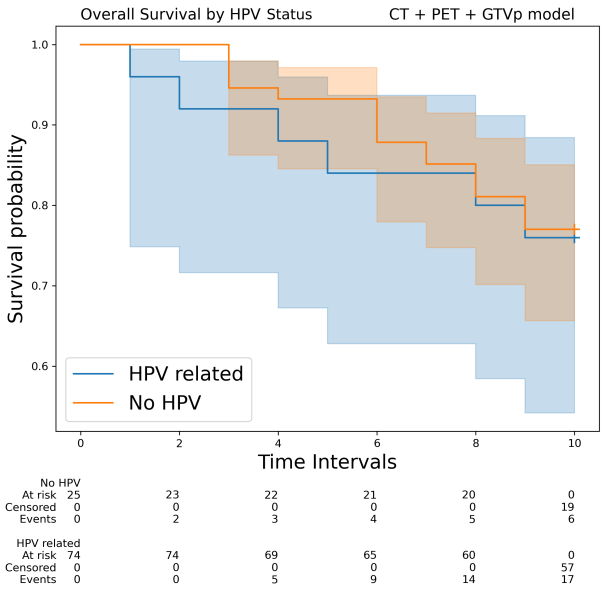
Figure 4.3 shows KM curves grouped by HPV status in oropharyngeal tumors for the OUS dataset and OS endpoint. Figure 4.3a shows that observed ground truth data, while Figure 4.3b shows the predicted survival probabilities by the CT+PET+GTVp model. The curves span the ten time intervals the model used for prediction, which is over a five year period where time interval 10 equals 60 months. The y-axes are different between the two panels Figure 4.3a and Figure 4.3b. Figure 4.3a ranges from 0.30 to 1.0, and Figure 4.3b ranges from 0.50 to 1.0. The KM curves in Figure 4.3a had the observed survival probabilities around 0.70 and 0.40 for *HPV related* tumors and *No HPV* relation respectively, at the end of time interval 10. The model estimated survival in Figure 4.3b had an estimated probability around 0.90 and 0.70 for *HPV related* tumors and *No HPV* relation respectively, at the last time interval. This shows that the model estimated higher survival than the observed ground truth. The number of patients that experienced the event were, for the observed ground truth, 27 patients with *HPV related* tumors, and 30 patients with *No HPV* relation. The model estimates were 8 events for patients with *HPV related* tumors, and 17 events for patients with *No HPV* relation, at time interval 10.

Figure 4.3a demonstrated a higher survival probability for patients with *HPV related* tumors over those with *No HPV* relation. The log-rank test results for the ground truth curves, shown in Table 4.5, gave a $p - value$ of under 0.005. This indicates a significant difference between the survival probabilities of patients with or without HPV related tumors.

The model predicted survival probabilities in Figure 4.3b showed a similar trend as the ground truth, where patients with *HPV related* tumors had a greater survival probability that those with *No HPV* relation. Until around time interval 4, corresponding to month 24, the KM curves and their confidence interval overlap, and the curves even crossed, indicating that the model had not differentiated between the survival probabilities of the two groups. The log-rank test in Table 4.5 gave a $p - value$ of 0.01. This is under the set threshold at 0.05 and therefore indicated a significant difference in predicted survival probability, but less so than for the observed ground truth.



(a) Observed ground truth for the MAASTRO data.



(b) Model prediction for the MAASTRO data.

Figure 4.4: KM curves grouped by HPV status for the OS endpoint using the CT+PET+GTVp model. The KM curves show the survival probability of two groups, the blue curve with *HPV related* tumors, and the orange curve with *No HPV* relation. The figure is divided into two panels, (a) shows the observed survival present in the MAASTRO dataset, called the ground truth, (b) shows the model estimated survival for the MAASTRO dataset. The shaded area around the curves represent the 95% confidence intervals, which indicates the precision of the survival estimate, the narrower the confidence interval, the more precise the estimate is. Note that the panels have different y-axes, (a) ranging from 0.30 to 1.0, and (b) ranging from 0.60 to 1.0. Under the KM curves in panel (a) and panel (b) are shown the number of patients at risk, and the events and censoring counts for each time interval.

Table 4.6: Log-rank test for the KM curves given in Figure 4.4.

Ground Truth MAASTRO			**Predicted MAASTRO**		
Test statistic	p-value	$-log_2(p)$	Test statistic	p-value	$-log_2(p)$
0.24	0.63	0.67	0.04	0.85	0.24

Figure 4.4 shows KM curves grouped by HPV status in oropharyngeal tumors on the external MAASTRO dataset and OS endpoint. Figure 4.4a and Figure 4.4b show the actual ground truth data and the predicted survival probability by the CT+PET+GTVp model. The curves span ten time intervals corresponding to a five-year period, where time interval 10 equals 60 months. Figure 4.4a and Figure 4.3b have different y-axes. Figure 4.4a ranges from 0.30 to 1.0, and Figure 4.4b ranges from 0.60 to 1.0. The KM curves in Figure 4.4a had the observed survival probabilities around 0.40 for both *HPV related* tumors and *No HPV* relation, at the end of time interval 10. The model estimated survival in Figure 4.4b had an estimated probability around 0.80 for both *HPV related* tumors and *No HPV* relation, at the last time interval. This shows that the model estimated higher survival than the observed ground truth. The number of patients that experienced the event were, for the observed ground truth, 12 patients with *HPV related* tumors, and 21 patients with *No HPV* relation. The model estimates were 5 events for patients with *HPV related* tumors, and 14 events for patients with *No HPV* relation, at time interval 10.

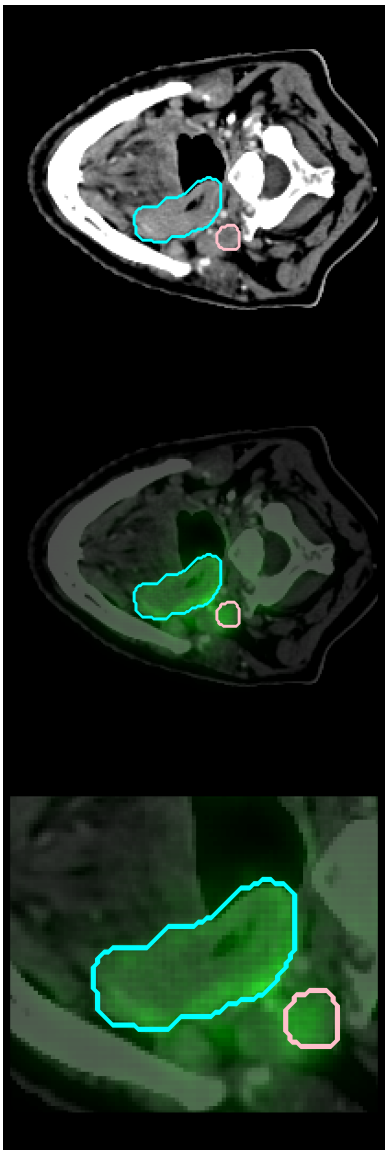
The ground truth in Figure 4.4a show overlapping KM curves across the entire follow-up period. The log-rank test in Table 4.6 gave a $p-value$ of 0.63, much higher than the 0.05 threshold for significance. This confirms that there were no significant differences between the survival probabilities for patients with *HPV related* tumors and patients with *No HPV* relation in the MAASTRO dataset.

Similarly, the model predicted survival probabilities, in Figure 4.4b, showed the same pattern of overlapping curves. The corresponding $p-value$ of 0.85 indicates that the model had not predicted a separation between the groups.

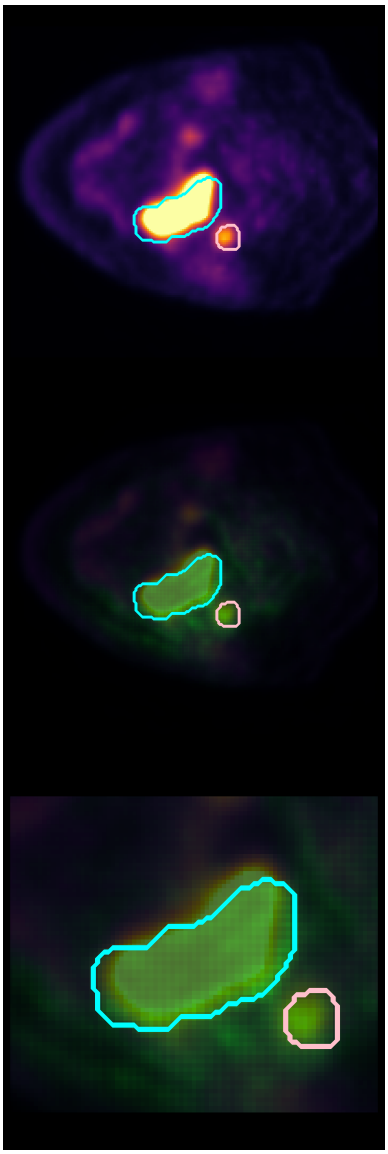
4.3 Explainability Methods

4.3.1 VarGrad Saliency Maps

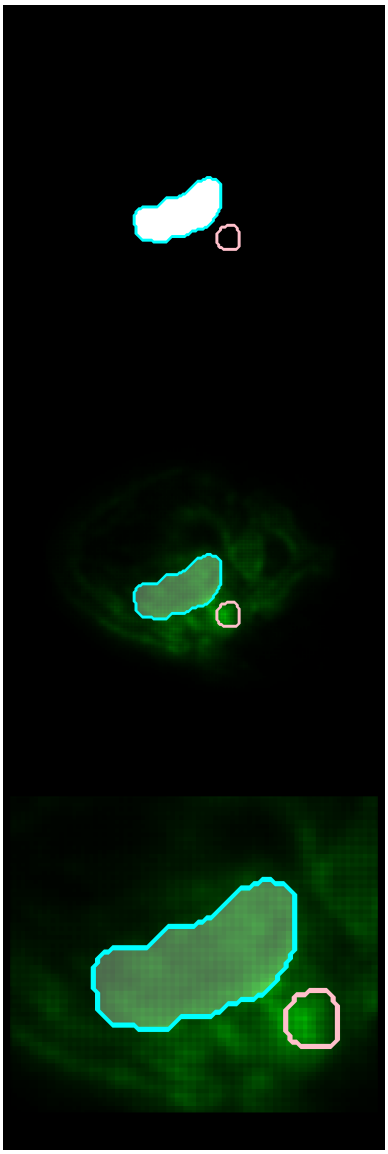
Overall Survival



(a) Original CT image, VarGrad for CT input channel and magnification of the VarGrad highlighted region.



(b) Original PET image, VarGrad for PET input channel and magnification of the VarGrad highlighted region.



(c) Original GTVp contour, VarGrad for GTVp input channel and magnification of the VarGrad highlighted region.

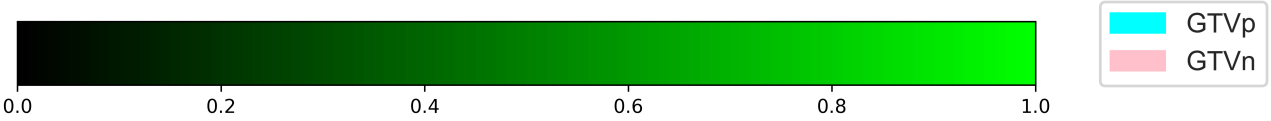


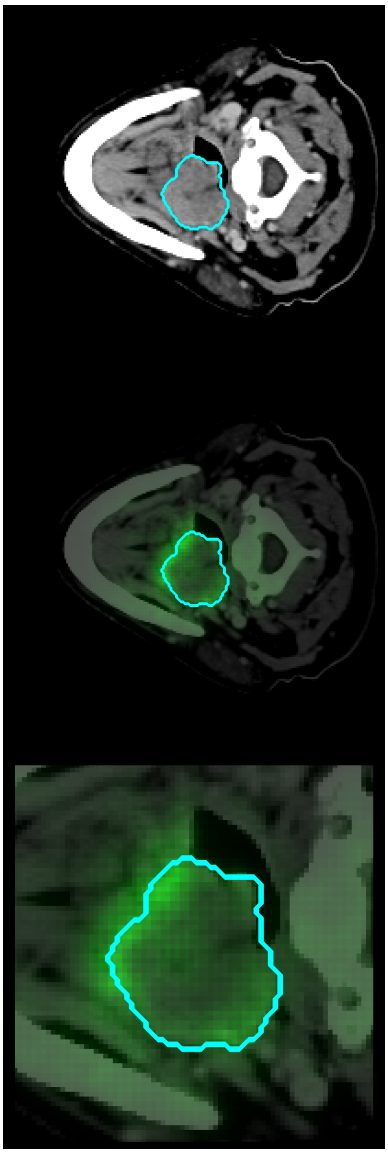
Figure 4.5: VarGrad heatmaps with the corresponding modalities on the OS endpoint for the OUS dataset, using the CT+PET+GTVp model. The VarGrad heatmaps, shown in green overlay over the input images, range in intensity from 0 to 1, as shown in the color bar, where a higher intensity means a higher significance for model predictions. The images come from one slice of one patient from the OUS dataset. The patient slice contains both a primary tumor and a nodal area, shown in the primary tumor, GTVp, and nodal, GTVn, overliad delineations. The first row for each input channel shows a slice of the unaltered input image given to the model. The second row shows the input image with overlaying VarGrad heatmap. The third row shows the overlaying VarGrad heatmap zoomed in on the region of interest.

VarGrad heatmaps in Figure 4.5 identify regions within the input image that the model considers significant for predicting survival probability. The GTVp and GTVn contours outlined on the images correspond to the actual tumor and nodal regions delineated by the oncologist, and were not given to the model unless shown in one of the figure panels as an input image. The CT+PET+GTVp model was not given the nodal contour as input, yet the VarGrad heatmap was shown highlighting this area for all input images.

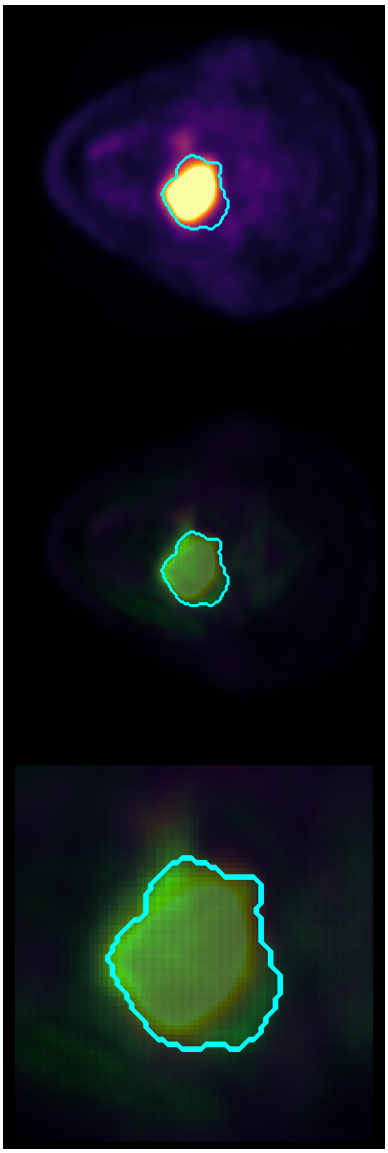
Figure 4.5a shows a slice of the VarGrad heatmap for the CT input channel. The heatmap shows the areas of the CT modality the model finds most important for prediction. The VarGrad heatmap highlighted the primary tumor and nodal regions of the CT image.

Figure 4.5b shows a PET slice with an overlaid VarGrad heatmap. The highest values of VarGrad were found in around the same region as the highest SUV values, which correspond to the primary tumor and nodal areas.

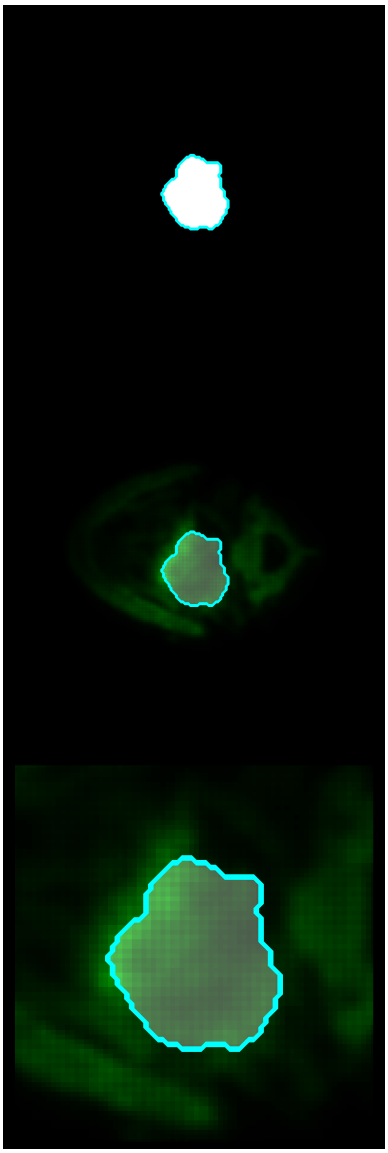
Figure 4.5c shows the VarGrad heatmap corresponding to the GTVp input channel. Here, the VarGrad highlighted the area delineated by the GTVp contour, while still mostly found around it.



(a) Original CT image, VarGrad for CT input channel and magnification of the VarGrad highlighted region.



(b) Original PET image, VarGrad for PET input channel and magnification of the VarGrad highlighted region.



(c) Original GTVp contour, VarGrad for GTVp input channel and magnification of the VarGrad highlighted region.

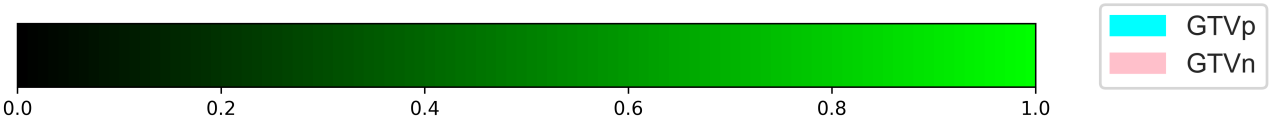
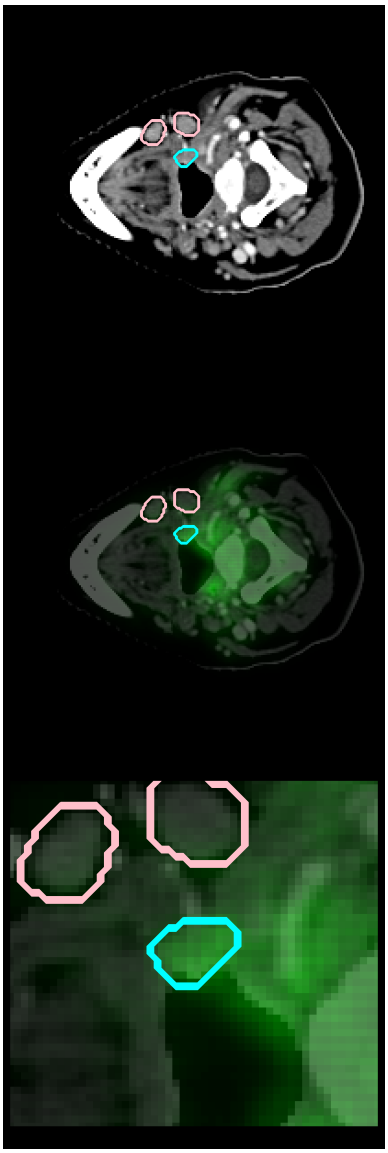


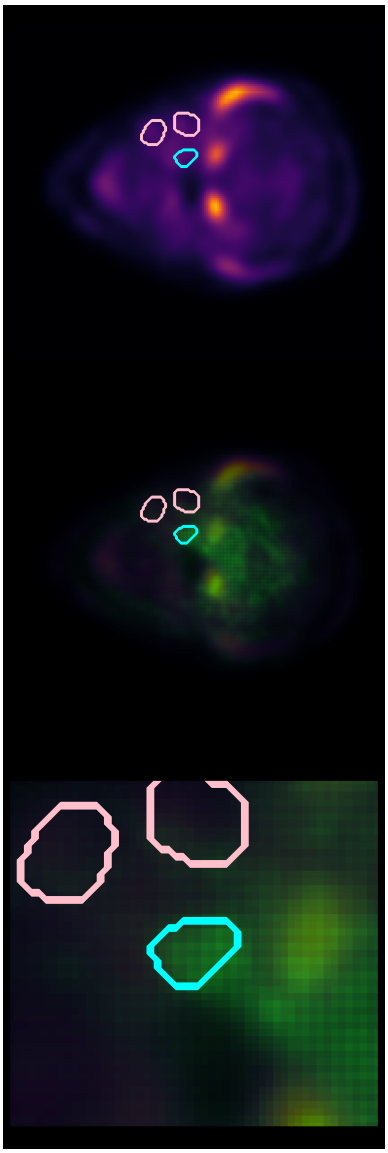
Figure 4.6: VarGrad heatmaps with the corresponding modalities on the OS endpoint for the external MAASTRO dataset, using the CT+PET+GTVp model. The VarGrad heatmaps, shown in green overlay over the input images, range in intensity from 0 to 1, as shown in the color bar, where a higher intensity means a higher significance for model predictions. The images come from one slice of one patient from the MAASTRO dataset. The patient slice only contains a primary tumor, shown in the GTVp delineation overlaid. The first row for each input channel shows a slice of the unaltered input image given to the model. The second row shows the input image with overlaying VarGrad heatmap. The third row shows the overlaying VarGrad zoomed in on the region of interest.

Figure 4.6 shows the VarGrad for the CT+PET+GTVp model predicting on the external MAASTRO dataset on the OS endpoint. Note that the patient had no nodal areas in the slices shown, only a primary tumor.

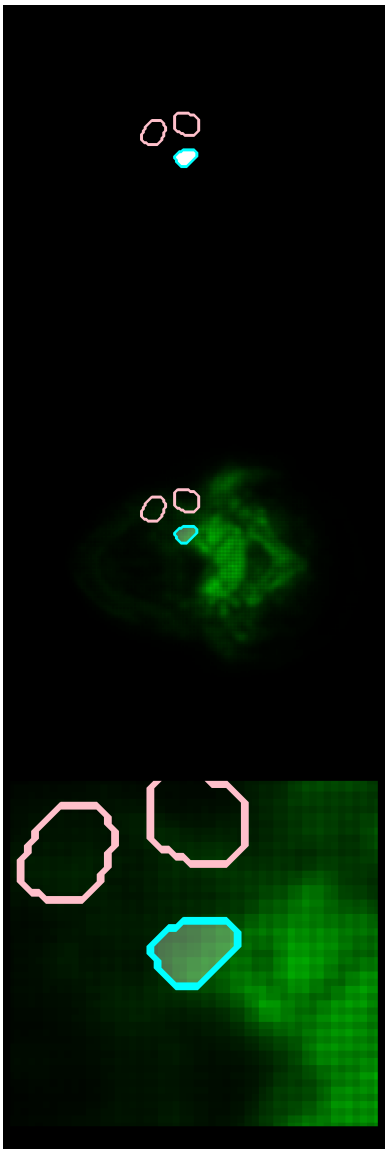
The VarGrad heatmap corresponding to the CT modality in Figure 4.6a was found mostly around the tumor area, but more in the periphery than inside. The VarGrad heatmap for the PET channel, seen in Figure 4.6b, overlapped the highest SUV areas, which themselves were found within the tumor area. The peak of the GTVp channel's VarGrad heatmap, seen in 4.6c, was found within the delineated primary tumor area, but as in Figure 4.5c it was also seen outside.



(a) Original CT image, VarGrad for CT input channel and magnification of the VarGrad highlighted region.



(b) Original PET image, VarGrad for PET input channel and magnification of the VarGrad highlighted region.



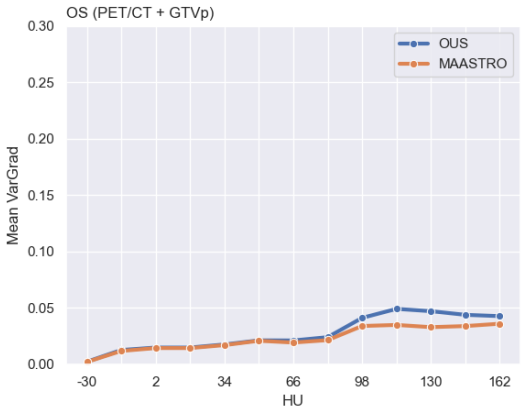
(c) Original GTVp contour, VarGrad for GTVp input channel and magnification of the VarGrad highlighted region.

Figure 4.7: VarGrad heatmaps with the corresponding modalities on the OS endpoint for the external MAASTRO dataset, using the CT+PET+GTVp model. The VarGrad heatmaps, shown in green overlay over the input images, range in intensity from 0 to 1, as shown in the color bar, where a higher intensity means a higher significance for model predictions. The images come from one slice of one patient from the MAASTRO dataset. The patient slice contains both a primary tumor and a nodal area, shown in the primary tumor, GTVp, and nodal, GTVn, delineations overlaid. Notably for this patient, the PET modality failed to show the tumor and nodal areas. The first row for each input channel shows a slice of the unaltered input image given to the model. The second row shows the input image with overlaying VarGrad heatmap. The third row shows the overlaying VarGrad zoomed in on the region of interest.

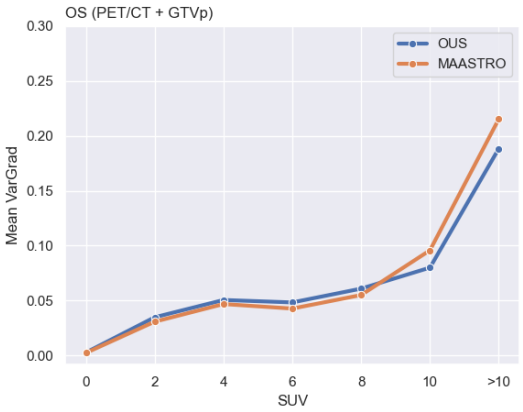
Figure 4.7 shows a case where the PET modality did not align with the tumor and nodal areas. This was because the patient had excessive use, or inflammation, of the sternocleidomastoid muscle before PET acquisition.

For this patient, the VarGrad heatmap highlighting was found mostly outside the primary tumor and nodal areas, while still highlighting the primary tumor, for all modalities and contours, as seen in Figure 4.7a, Figure 4.7b and Figure 4.7c.

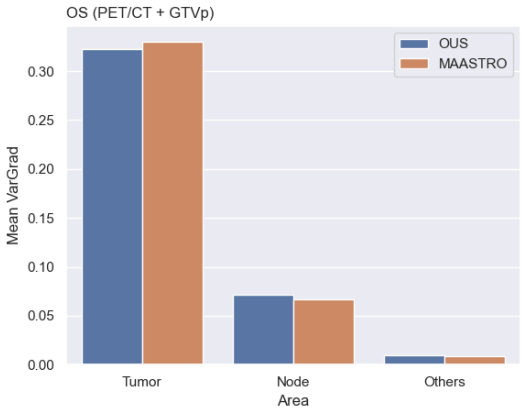
Overall Survival Statistical Plots



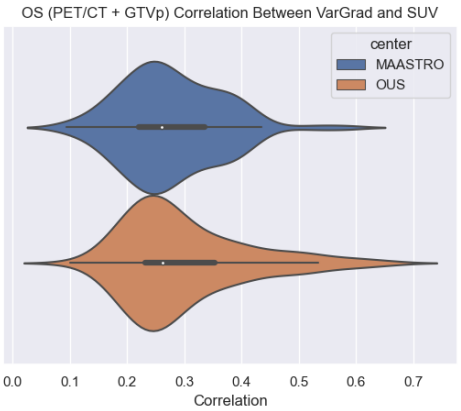
(a) Mean VarGrad values against HU of the CT modality.



(b) Mean VarGrad values against SUV values of the PET modality.



(c) Mean VarGrad values per region.



(d) Correlation between VarGrad values and SUV per patient.

Figure 4.8: VarGrad statistical plots for the OS endpoint based on the CT+PET+GTVp model. The panels show plots for both the OUS and external MAASTRO datasets, in blue and orange, respectively. Panel (a) shows the mean VarGrad values plotted against HU of the CT modality, giving an indication of the correlation of VarGrad values to the values of the CT images. (b) shows the mean VarGrad values plotted against SUV of the PET modality, giving an indication of the correlation of VarGrad values to the values of the PET images. (c) shows which regions the VarGrad heatmap highlighted. The regions are the primary tumor area, called *Tumor*, the nodal areas, called *Node*, and the region outside those deliniations, called *Others*. (d) shows the distribution of each patient's mean VarGrad value's correlation with the SUV values from that patient's PET image. The plot is a violin plot, showing the distribution, with an inner box plot, showing the correlation interquartile range and median correlation coefficient.

Figure 4.8 shows a series of statistical plots for the VarGrad analysis of the CT+PET+GTVp model predicting on the OS endpoint. The plots show data for model predictions on all patients on both the OUS and external MAASTRO datasets.

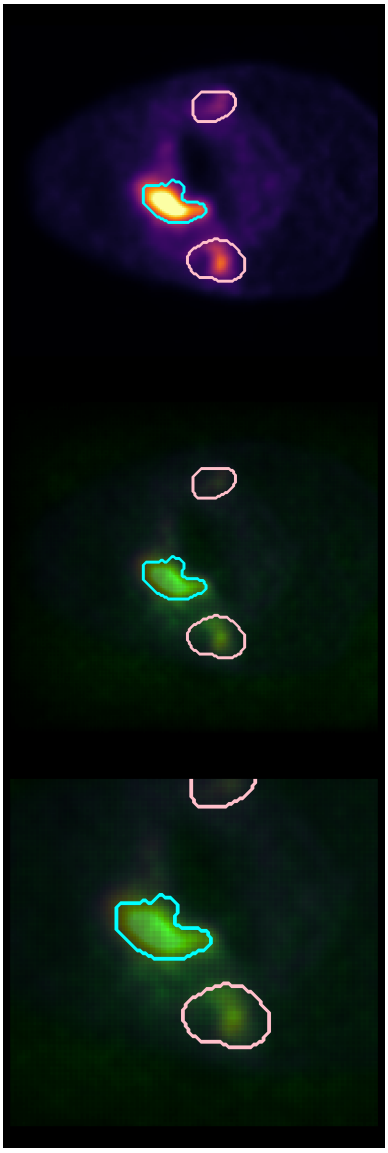
The first plot, Figure 4.8a, shows the mean VarGrad value against HU values of the CT modality. The plot indicates what values in the CT modality correspond to mean VarGrad values. As HU increased, the mean VarGrad values were relatively stable, with an increase in mean VarGrad value from around 90 HU, for both datasets. The mean VarGrad values ranged from 0 to a max mean VarGrad value around 0.05 across all HUs.

The second plot, Figure 4.8b, shows the mean VarGrad values against values of SUV of the PET modality. The plot shows that higher values in SUV corresponded to a higher mean VarGrad, with a peak at SUV > 10. Unlike in Figure 4.8a, where the mean VarGrad was relatively stable over values of HU, the mean VarGrad ranged from 0 to a max mean VarGrad around 0.22, over all values of SUV. Results for both datasets followed the same trend.

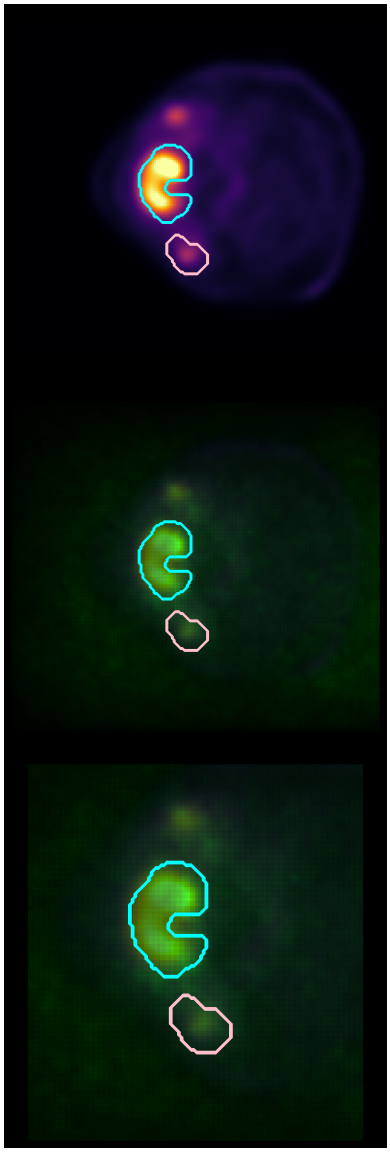
The third plot, Figure 4.8c, shows mean VarGrad values within each delineated area: primary tumor, called *Tumor*, nodal regions, called *Node*, and *Other* regions outside of the provided contours. The plot shows that mean VarGrad values were mostly present within the primary tumor area, and somewhat present within the nodal areas. It was almost not found in the *Other* area. Both datasets followed the same pattern.

The fourth plot, Figure 4.8d, shows the correlation between VarGrad values and SUV for all patients in both datasets. The plot shows the distribution of correlation coefficients for both datasets, with an inner box plot showing the interquartile range and median correlation coefficient. Both datasets had a wide distribution, meaning that some patients showed a strong relationship between VarGrad values and SUV, while others showed a weaker relationship. Both datasets had around the same median correlation coefficient of 0.25.

Disease Free Survival



(a) Original PET image, VarGrad for PET input channel and magnification of the VarGrad highlighted region, for the OUS dataset.



(b) Original PET image, VarGrad for PET input channel and magnification of the VarGrad highlighted region, for the MAASTRO dataset.

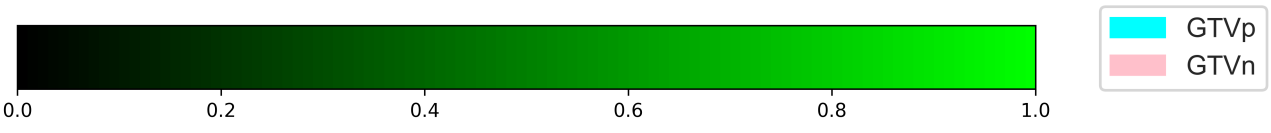
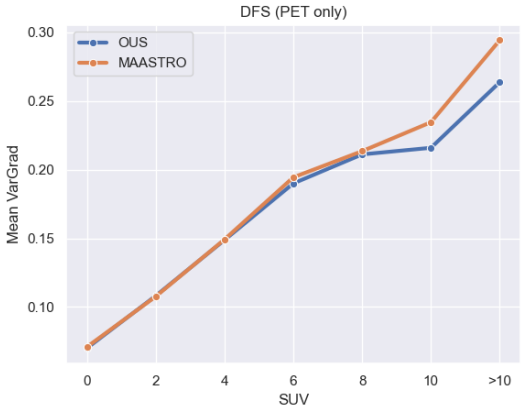


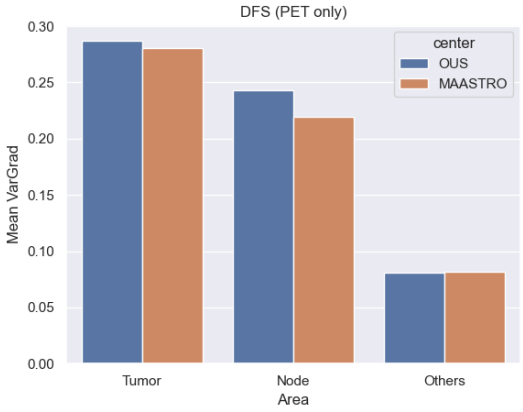
Figure 4.9: VarGrad heatmaps with the corresponding modality on the DFS endpoint for both the OUS and external MAASTRO datasets, using the PET-only model. The VarGrad heatmaps, shown in green overlay over the input images, range in intensity from 0 to 1, as shown in the color bar, where a higher intensity means a higher significance for model predictions. The two panels show images from one slice of one patient from each of the datasets. Panel (a) shows a patient from the OUS dataset. Panel (b) shows a patient from the MAASTRO dataset. The patient slices contain both a primary tumor and a nodal areas, shown in the primary tumor, GTVp, and nodal, GTVn, overlaid delineations. The first row for each input channel shows a slice of the unaltered input image given to the model. The second row shows the input image with overlaying VarGrad heatmap. The third row shows the overlaying VarGrad zoomed in on the region of interest.

Figure 4.9 shows VarGrad heatmaps for the PET-only model predicting the DFS endpoint on both datasets. Figure 4.9a shows the VarGrad results for the model predictions on the OUS dataset, and Figure 4.9b on the MAASTRO dataset. Both VarGrad heatmaps are concentrated around the areas with the highest SUV, which roughly corresponded to the tumor and nodal areas.

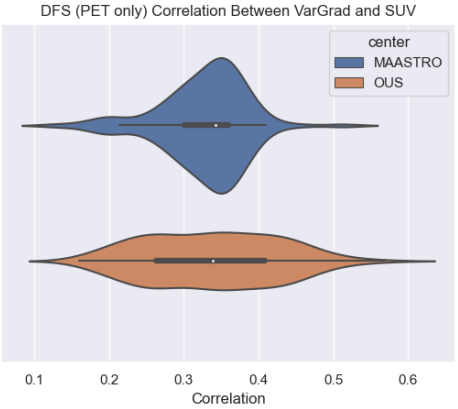
Disease Free Survival Statistical Plots



(a) Mean VarGrad against SUV.



(b) Mean VarGrad per region on DFS for PET.



(c) Correlation between VarGrad and SUV per patient on DFS for PET.

Figure 4.10: VarGrad statistical plots for the DFS endpoint based on the PET-only model. The panels show plots for both the OUS and external MAASTRO datasets, in blue and orange, respectively. Panel (a) shows the mean VarGrad values plotted against SUV of the PET modality, giving an indication of the correlation of VarGrad values to the values of the PET images. (b) shows which regions the VarGrad heatmap highlighted. The regions are the primary tumor area, called *Tumor*, the nodal areas, called *Node*, and the region outside those deliniations, called *Others*. (c) shows the distribution of each patient's mean VarGrad value's correlation with the SUV value from that patient's PET image. The plot is a violin plot, showing the distribution, with an inner box plot, showing the correlation interquartile range and median correlation coefficient.

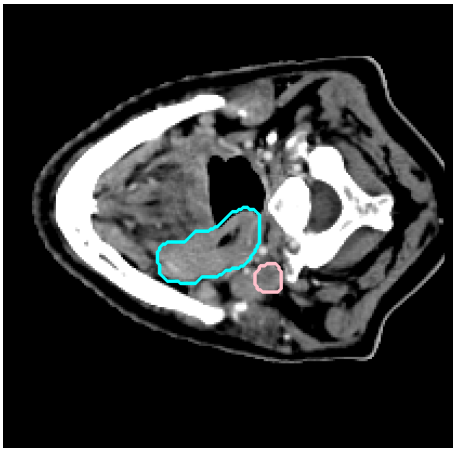
Figure 4.10 shows a series of statistical plots for the PET-only model predicting on the DFS endpoint for both OUS and MAASTRO datasets.

The first plot in Figure 4.10a shows the relationship between mean values of VarGrad and SUV. The graph shows a trend where increased SUV values lead to an increase in mean VarGrad. The VarGrad values ranged from 0 to a max SUV of 0.30 over all values of SUV. This pattern was shared between the datasets.

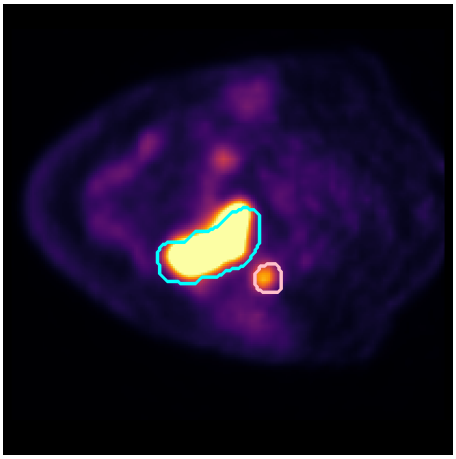
The second plot, Figure 4.10b, shows VarGrad values across different regions. Here, the VarGrad values were found mostly within the tumor and nodal areas, where most were within the primary tumor for both datasets. Some VarGrad highlighting was also found outside these areas.

The violin plot in Figure 4.10c shows the correlation between mean VarGrad values and SUV for all patients. The distributions show a span of correlations among patients. Patients in the MAASTRO dataset were less spread out than in the OUS dataset. This means more variability in how the VarGrad values correlates with SUV in the OUS dataset than in the MAASTRO dataset.

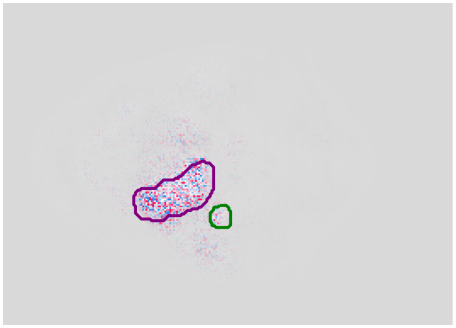
4.3.2 SHAP Saliency Maps



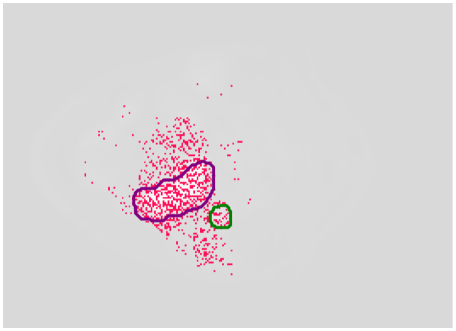
(a) Original CT image with primary tumor contour.



(b) Original PET image with primary tumor contour.



(c) Raw Gradient SHAP values overlaid on the PET image.



(d) Thresholded Gradient SHAP values overlaid on the PET image.

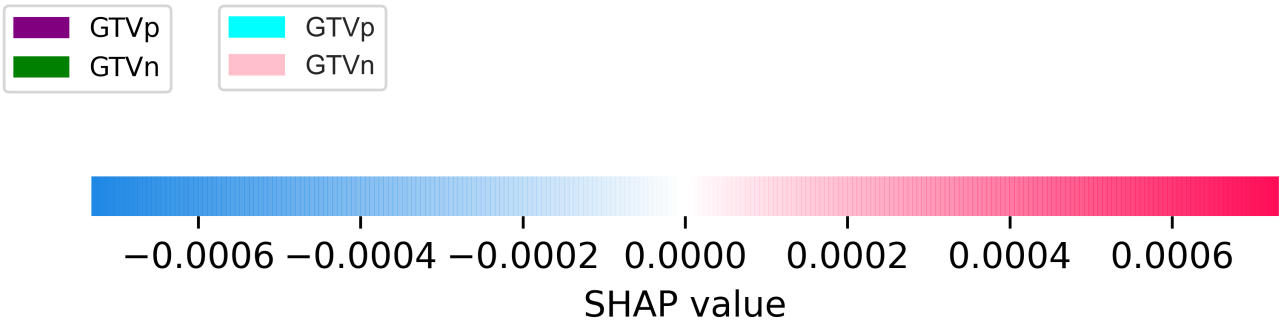


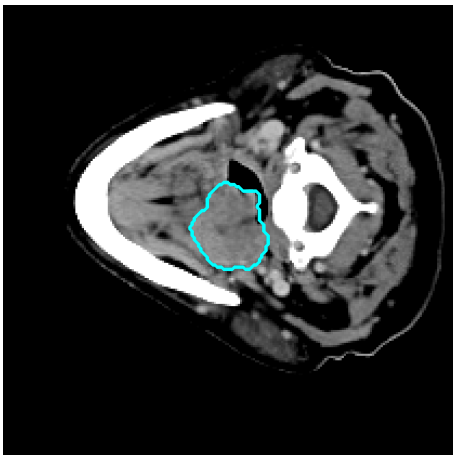
Figure 4.11: Gradient SHAP values computed with the PET-only model for one patient from the OUS dataset on the OS endpoint. The SHAP values are displayed as one slice from one patient for the 9^{th} time interval. The patient slice contains a primary tumor and nodal areas, delineated by the GTV contours. The primary tumor contour, GTVp, is shown in cyan in (a) and (b), and in purple in (c) and (d). The nodal area contour, GTVn, is shown in pink in (a) and (b), and in green in (c) and (d). The image is the same slice as used for the VarGrad images in Figure 4.5. Under the images is shown a color bar for the SHAP value range. This color bar only applies to the raw Gradient SHAP values in panel (c). (a) shows the original CT image of the slice with GTVp contour. The model was not given CT images as input. (b) shows the original PET image of the slice with GTVp contour. (c) shows the raw Gradient SHAP values as computed by the *GradientSHAP* explainer using the PET-only model, with GTV contours overlaid. The SHAP values are represented as colored regions of the image, each color representing a SHAP value. The corresponding color bar shows which color corresponds to which SHAP values. Negative SHAP values, shown in blue, correspond to a negative prediction, here, prediction the event occurred within the interval. Positive SHAP values, shown in red, represent points that correlate to a positive prediction, here, predicting survival through the time interval. (d) shows Gradient SHAP values that were thresholded to be the 1% most significant values for model prediction. The values are binary, 1 if they were the 1% most significant values for model prediction, 0 if not. The color of the binary thresholded SHAP values is insignificant.

Figure 4.11 shows raw and thresholded Gradient SHAP values and the corresponding original CT and PET images. The model used to compute the SHAP values was the PET-only model for the OUS dataset on the OS endpoint. This is the same slice as used in the VarGrad image in Figure 4.5. The values are shown for time interval 9.

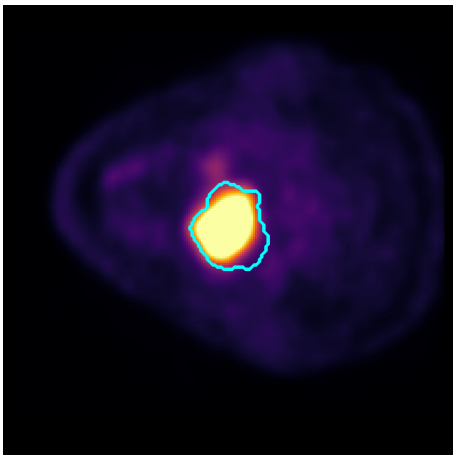
Figure 4.11a and Figure 4.11b show the original CT and PET image slices corresponding to the slices of raw and thresholded SHAP values. Since the model used to compute the SHAP values was a PET-only model, the model was not given the CT image as input, only the PET image.

Figure 4.11c shows the raw Gradient SHAP values. The values are shown with a primary tumor, GTVp, and nodal area, GTVn, contour. The SHAP values are represented as colored regions of the image, each color representing a SHAP value. The intensity of the colors shows how significant that SHAP value was for model predictions. The raw SHAP values are a mix of positive and negative, mostly concentrated within the tumor and node area. Negative SHAP values, shown in blue, correspond to a negative prediction, here, prediction the event occurred within the interval. Positive SHAP values, shown in red, represent points that correlate to a positive prediction, here, predicting survival through the time interval. There's no clear preference for one color in the raw Gradient SHAP values.

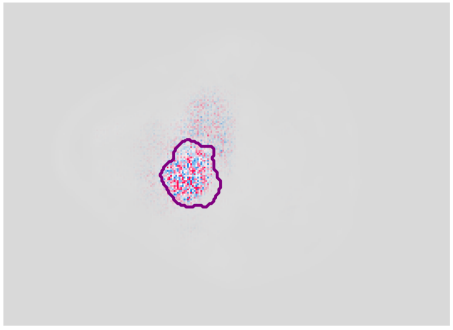
Figure 4.11d shows Gradient SHAP values thresholded to be the 1% most significant values for model prediction. The thresholded SHAP values are binary, and therefore, the color is irrelevant. The amount of points is fewer than the raw SHAP output of Figure 4.11c. The thresholded Gradient SHAP values are concentrated around the tumor and node areas, but are also found outside the delineations.



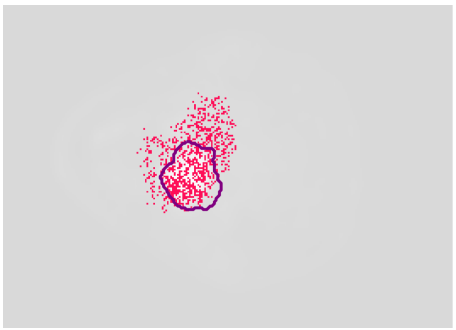
(a) Original CT image with primary tumor contour.



(b) Original PET image with primary tumor contour.



(c) Raw Gradient SHAP values overlaid on the PET image.



(d) Thresholded Gradient SHAP values overlaid on the PET image.

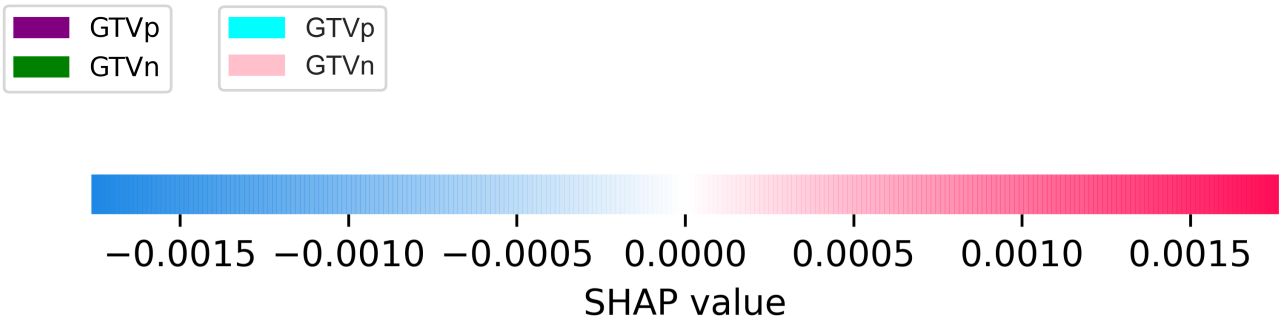


Figure 4.12: Gradient SHAP values computed with the PET-only model for one patient from the external MAASTRO dataset on the OS endpoint. The SHAP values are displayed as one slice from one patient for the 9^{th} time interval. The patient slice only contains a primary tumor, delineated by the GTVp contour. he primary tumor contour, GTVp, is shown in cyan in (a) and (b), and in purple in (c) and (d). The image is the same slice as used for the VarGrad images in Figure 4.6. Under the images is shown a color bar for the SHAP value range. This color bar only applies to the raw Gradient SHAP values in panel (c). (a) shows the original CT image of the slice with GTVp contour. The model was not given CT images as input. (b) shows the original PET image of the slice with GTVp contour. (c) shows the raw Gradient SHAP values as computed by the *GradientSHAP* explainer using the PET-only model, with GTV contours overlaid. he SHAP values are represented as colored regions of the image, each color representing a SHAP value. The corresponding color bar shows which color corresponds to which SHAP values. Negative SHAP values, shown in blue, correspond to a negative prediction, here, prediction the event occurred within the interval. Positive SHAP values, shown in red, represent points that correlate to a positive prediction, here, predicting survival through the time interval. (d) shows Gradient SHAP values that were thresholded to be the 1% most significant values for model prediction. The values are binary, 1 if they were the 1% most significant values for model prediction, 0 if not. The color of the binary thresholded SHAP values is insignificant.

Figure 4.12 shows raw and thresholded Gradient SHAP values computed with a PET-only model on the OS endpoint using the external MAASTRO dataset. This is the same slice as used in the VarGrad in Figure 4.6. The values are shown for time interval 9.

Figure 4.12a and Figure 4.12b show the original CT and PET image slices corresponding to the slices of raw and thresholded SHAP values. Since the model used to compute the SHAP values was a PET-only model, the model was not given the CT image as input, only the PET image.

The strongest colored raw SHAP values in Figure 4.12c, and therefore the most significant, seem to be inside the primary tumor area, while more dim values are found just outside it.

Figure 4.12d shows SHAP values thresholded to the 1% most significant values. Most thresholded SHAP values are found in the primary tumor area, while some are found outside around it. The color of the thresholded SHAP values is irrelevant, since the thresholded SHAP values were binarized.

Chapter 5

Discussion

5.1 Choice of Intervals

In [37], Gensheimer et al. used between 15 and 40 intervals that were more spaced out with increasing follow-up time and included around the same number of event occurrences in each time interval to limit model bias. To avoid bias in estimates of survival distributions, [65] suggests splitting the follow-up period into at least ten intervals.

The follow-up time in this thesis was split into ten intervals for all models except one. The choice of ten time intervals was found in preexisting literature [16] [65]. Selecting intervals from 6 months up to 60 months aims to capture the nuances in survival times in the earlier, more critical periods of post-diagnosis. The first few months after diagnosis or treatment are the most critical, and survival rates change more dramatically during this period [51] [16]. A higher number of intervals in the early stages of the follow-up period relative to the number of intervals allow for a more detailed analysis of survival probabilities when they are most variable. This approach can increase the resolution of the analysis, especially in the early follow-up period where most events occur. In the dataset there are a higher density of data points in the initial months with fewer events recorded as time progresses.

A PET-only model on the OS endpoint was run to assess the difference in setting the time intervals to 20, each interval then including about the same number of patients. These intervals were spaced up to month 90, giving a larger period than the ten intervals, and therefore focusing less on the beggining of the follow-up period. The PET-only model using the 20 time intervals achieved a C-index of 0.69. This was worse than the PET-only model using ten time intervals which had a C-index of 0.70. Due to this small difference, and the fact that a higher number of intervals could make the analysis overly complex, or the intervals too sparsely populated with events, the choice was made to only test models on the 10-interval split. This is in line with Gensheimer et al. [37] that stated “In our experiments we have found that the model’s performance is fairly robust to choice of specific cut-points”.

5.2 Model Assessment

The C-index is the primary metric used to evaluate the performance of the models. This is because, unlike AUC, the C-index takes into account censoring information [42]. IBS shows the model's error in predictions rather than the correctness of the order of predictions [44]. C-index is preferred to IBS because it provides a direct measure of the model's ability to discriminate between different individuals in terms of risk. Other studies often only report the C-index, and therefore are only comparable to this thesis on that metric [16] [18] [17]. The AUC and IBS metrics can be more informative in certain contexts. The AUC can give the performance of the model at a specific instance of time, rather than only over the entire follow-up period. The IBS gives both the accuracy of the predicted probabilities and the timing of the events. It gives a combined measure of how close the predicted probabilities are to actual probabilities and how well the model distinguishes between events and non-events across all time points.

5.2.1 OS prognosis

For the OS prediction on the OUS dataset, the CT+PET+GTVp model emerged as the leading model with a C-index of 0.74, AUC of 0.69 and IBS of 0.16. The model had the highest C-index and AUC and the lowest IBS of any other OUS OS model. This model integrated CT, PET, and a GTV contour of the primary tumor, implying that the combination of imaging modalities with primary contour provided a comprehensive approach to predicting patient survival. It notably outperformed the single-modality models (CT and PET only), emphasizing the added value of multi-modal data integration. It also outperformed the CT+PET model without the GTVp contour, which had a C-index of 0.66. The worst model was the CT-only model, which had a C-index of 0.61, AUC of 0.56 and IBS of 0.18. This, together with the fact that models with the CT modality generally performed lower than models without CT, suggests that the CT modality, at least by itself, may not be well suited for time-to-event predictions.

For the external MAASTRO dataset, the CT+PET+GTVp+GTVn model had the highest C-index of 0.69, again reinforcing the notion that an approach utilizing multiple modalities provides a superior predictive capability. It is important to note that the highest performing model on the OUS dataset, the CT+PET+GTVp model, showed a lower C-index of 0.68 in the MAASTRO dataset, a decrease of 0.06. This decrease might be attributed to variations in the external dataset or differences in the patient populations, suggesting the model was potentially overfitting to the training dataset. The CT+PET+GTVp+GTVn model outperformed the CT+PET model with no GTV contours, which had a C-index score of 0.63. On the other hand, the PET-only model performed close to the best model, with a C-index of 0.67, suggesting that GTV contours might not be essential to high model performance. The CT-only model was the worst performing model on all metrics with a C-index of 0.62, AUC of 0.60 and IBS of 0.18.

5.2.2 DFS prognosis

When evaluating DFS, the model performances were generally lower than those for OS. The IBS scores of the DFS models using the OUS dataset ranged from 0.23 to 0.24. These IBS scores were close to the expected score from a randomly guessing model, which is 0.25. While the IBS showed a similar low performance for all DFS models on the OUS dataset, the C-index and AUC scores could differentiate them. In the OUS dataset, the PET only and PET+GTVp models scored the highest on the C-index, with the scores 0.62 and 0.60 respectively. This suggests that for DFS predictions, the CT modality did not provide useful information and contributed some noise to the model, leading to a lower performance. The CT-only model performed the worst on all metrics with a C-index of 0.51, AUC of 0.49 and IBS of 0.24. This score is close to a model making random guesses, which would have the expected performance of a C-index score of 0.50, AUC of 0.50 and IBS of 0.25.

All DFS models had an increase in C-index scores and a decrease in IBS error when evaluated on the external MAASTRO dataset. This suggests that the models may be more capable of predicting DFS in the context of the MAASTRO dataset rather than in the OUS dataset. As seen in Table 3.1, the MAASTRO dataset has different characteristics compared to the OUS dataset which potentially could make the DFS event outcomes more distinguishable by the models. The DFS event distribution could also be different in the MAASTRO dataset, potentially having a clearer separation between patients with events and those without. This could enhance the model's predictive performance. Another explanation is that the DFS models were more generalizable and not overfitted to the OUS training data, having captured underlying patterns that were valid across different populations. The PET-only model achieved the highest C-index score of 0.67, a gain of 0.05 from the OUS prediction. Notably, the CT-only model increased the C-index score with 0.13 to a score of 0.64, making it the third highest performing model on the C-index.

5.2.3 Modality Importance

The overall highest performing model, using the C-index, for predicting OS and DFS was the CT+PET+GTVp model. The robust performance of this model can be linked to its ability to capture both anatomical and functional information from CT and PET images, combined with the insights provided by the primary tumor volume contour in GTVp. However, focusing on the C-index, the model's performance was only slightly better than the other MAASTRO models.

The consistency of the multi-modal modeling approach in providing higher C-index values in both datasets implies that the complexity of cancer prognosis may be better navigated with a rich set of diverse modalities.

With exception of the highest performing model, the CT+PET+GTVp model, the performance of models incorporating CT images for predicting both OS and DFS in both the OUS and MAASTRO datasets can be seen to occupy the lower end of the performance spectrum when judged by the C-index. This suggests that models relying solely on CT or in combination with fewer modalities are less effective at capturing the complexity of cancer prognosis compared to those integrating more diverse data types.

In the OUS dataset for OS prediction, the CT-only model had a C-index of 0.61, which was the lowest among the models evaluated. For DFS prediction on the same dataset, the single-modality CT model again ranked lowest with a C-index of 0.62. This consistent pattern of lower performance of CT-only models across both endpoints suggests that anatomical imaging alone may not be sufficient for time-to-event analysis. CT images lack the ability to capture metabolic activity [8] like PET images [10], and a simple delineation of the cancer like the GTV contours [11]. This could be to be critical for assessing survival prognosis. The higher performance of multi-modal models that included PET and GTV contour parameters alongside CT, suggests that the integration of metabolic and functional information alongside the anatomical data leads to a more holistic and thus more accurate prediction, at least on the OS endpoint.

Predictions on the DFS endpoint showed a preference for the PET-only model on both datasets. This suggests that while a multi-modal approach could give more accurate predictions on the OS endpoint, predictions on the DFS endpoint could be better with the use of a simple single-modality model.

When looking at the MAASTRO dataset, the CT+PET model, which included the anatomical imaging from CT, as well as the metabolic information from PET, performs relatively better than the more complex CT+PET+GTVp+GTVn model for DFS prediction, albeit only slightly with a C-index of 0.65 compared to 0.63. This, and CT being a part of the highest performing model on the C-index, could indicate that in the context of an external dataset with different patient characteristics, the anatomical information of a CT scan can help predicting the prognosis.

5.2.4 Model Robustness

Looking at the ensemble average model performances before vertically stacking and averaging, shown in Appendix B, the external results had a similar C-index even though the models were trained on different data splits. For example, the CT-only model trained on the OS endpoint in the MAASTRO dataset had very similar C-index for MAASTRO folds 0-4, even though fold 0 was trained and validated on OUS fold 1-4, while MAASTRO fold 1 was trained and validated on OUS fold 2-4. So even though each model had not seen a part of the OUS dataset, their performance on the external MAASTRO data was quite similar. This suggest that the models were stable, since they did not differ much in performance when exposed to new data from unseen folds.

5.2.5 Comparison With Other Studies

In [16], Wang et al. reported the highest performance with a PET-only model for both DM and OS predictions, achieving a C-index of 0.82 for the DM endpoint, and 0.69 for the OS endpoint. Their findings indicate that the metabolic imaging data from PET images could be the most important model input for the time-to-event models, without the necessity of the anatomical data from CT or GTV delineations.

Models relying on CT imaging scored, with exception of the highest performing model, lower on C-index in this thesis. This is consistent with the results from Wang et al. [16], where PET-based models outperformed those based on CT imaging. This reinforces the notion that the anatomical data from CT images could be less predictive of patient survival compared to the metabolic imaging provided by PET. While multi-modality approaches can be beneficial for certain tasks, they may reach a point of diminishing returns. Especially adding the extra information provided by the CT modality may only contribute noise to the model and reduce it's predictive performance.

Although the results in this thesis highlight the potential for multi-modality models, Wang et al. [16] found that single modality PET-only models perform the highest on both the OS and DM endpoints. The PET-only model achieved the highest C-index for DFS in this thesis. Together with the results from Wang et al. it is likely that a single-modality PET-only based approach could be more practical and equally robust for certain endpoints, like the DFS endpoint.

Wang et al. [16] also concluded that the GTV contour may be less relevant for models focusing on the PET modality, a finding not supported by this thesis. Notably, the GTV modality in Wang et al. combined both the GTVp and GTVn contours, which were split into two separate contours in the models of this thesis. Models including both GTV contours in this thesis did not perform as well as models with only the GTVp contour. This could explain the discrepancy between the model performances with the GTV contour in the study by Wang et al. and the model performances with GTVp contours in this thesis.

In [51], Moan et al. conducted a study that looked at the significance of FDG-PET parameters on DFS prediction, using univariate and multivariate Cox regression models. The study utilized the same data as the OUS datset used in this thesis. Moan et al. found that FDG-PET parameters, like SUV and metabolic tumor volume, were not significant predictors of the DFS endpoint. Especially for patients with HPV-related cancer, the GTV segmented area from the CT images was found to be more influential on predictions than the PET-related parameters. This is in contrast to the findings in this thesis, which found the PET modality to generally be the most influential model input. Particularly for predictions on the DFS endpoint, models relying the PET modality scored the highest on the C-index. The best model for both the OUS and external MAASTRO datasets when predicting on the DFS endpoint were the PET-only models. While the GTVn contour was not included in any model prediction on the DFS endpoint, the GTVp contour was part of the second and third highest C-index scoring model using the OUS dataset. The GTV contours in this thesis, however, were given as masks for both CT and PET, and not just a CT-based tumor volume like in Moan et al. Therefore, the importance of the GTV contours are not directly comparable between the studies.

The differences in importance of PET-related features when predicting on the DFS endpoint between the studies could be due to the models and methods. Moan et al. used a radiomics approach for feature extraction with Cox regression models for predictions [51], while this thesis used CNNs to automate the feature extraction and make predictions. The study by Moan et al. was also not making survival time predictions, like this thesis. Rather, Moan et al. used a Cox regression analysis of the different radiomics parameters. This difference in approach could explain the differing results of feature importance.

In [19], Rebaud et al. made a time-to-event radiomics model for the HECKTOR 2022 challenge. The model was predicting the time until recurrence free survival, which was defined as the time until a reappearance of a lesion or a new lesion. The model extracted 93 radiomics features from segmented tumor and nodal areas of CT and PET images, combining them with clinical features. The predictions were averages of ensemble models, using randomly chosen subsets of the training data and model features. The best model achieved a C-index of 0.68.

The endpoint predicted on in [19], being recurrence free survival, made the models not directly comparable to this thesis, which predicted on the endpoints OS and DFS. However, the features found to be important in the study by Rebaud et al. offer parallels to the feature importances found in this thesis. Rebaud et al. found a number of clinical features to have an impact on predictions, like tobacco usage. Of the radiomics features, large primary tumor diameter, high SUV in lesions and the number of affected nodal areas were found to be most important for model predictions. This supports the findings in this thesis, which found that the GTVp tumor area and PET modality could be particularly important for time-to-event predictions.

5.3 Kaplan-Meier Curves

5.3.1 Overall Stage of Disease

Figure 4.1 and Figure 4.2 show the true and estimated survival probabilities for the patients grouped by stages I-II and III-IV of disease. The log-rank tests, given in Table 4.3 and Table 4.4, confirm a statistically significant difference in survival probability between the two stage groups. Both the actual and predicted results showed p-values well below the chosen threshold of 0.05, indicating that stage of disease is a critical factor in patient prognosis, and is picked up by the model. The model was not given any clinical information of the stages of the cancer, and any separation of the stage groups found by the model is therefore predicted from the CT, PET and GTV inputs given to the model.

The ground truth for both the OUS and MAASTRO datasets show a consistent trend where lower stages of cancer had higher survival probabilities. Predictions for both datasets had a higher survival probability than the ground truth for both groups of stages. For example, at the end of the last time interval, looking at the *Stage III-IV* group on the OUS dataset, the observed data showed a survival probability of 0.30, while the model predicted 0.50. Likewise the observed data for the MAASTRO dataset, looking at the *Stage III-IV* group, showed a survival probability of 0.20 for the *Stage III-IV* group, while the model predicted 0.60. This consistent trend indicates that the model may overestimate the survival probability of all patients, regardless of stage or dataset.

5.3.2 HPV Positive Opharyngeal Tumors

In Figure 4.3 the KM prediction curves indicated an inability to separate survival probabilities based on HPV status in the first four time intervals, which correspond to the first 18 months. It was only after the fourth time interval that a divergence appeared. This could be the result of a time-dependent effect of HPV on survival. The effect of HPV on survival may become more pronounced over time as the disease progresses. This shows the importance of focusing the model on the initial months of the follow-up period where differences in survival may be harder to detect, and justifies the choice of splitting the follow-up period into ten intervals which has a higher proportion of intervals in the early period.

The MAASTRO dataset, as seen in Figure 4.4, did not show a significant difference in survival based on HPV status, with a log-rank p-value of 0.63, as seen in Table 4.6. The model predictions therefore did not significantly differ between the groups. Looking at the patient characteristics in Table 3.1, there was a lower proportion of HPV-related cancers in the MAASTRO dataset, only 22.2% compared to 57.6% in the OUS dataset. The smaller proportion of HPV-positive cases in the MAASTRO data could explain the lack of significant difference between the groups, compared to the OUS data where such cases were more prevalent. Furthermore, the MAASTRO dataset had a significantly higher proportion of patients with cancer stage III-IV, 80.8% compared to 48.2% for OUS, and packs of cigarettes smoked per year, a median of 40 packs per year compared to 22.5 for OUS. Given that advanced stages and tobacco usage are strongly associated with poorer survival [6], they may mask the influence of HPV status in this dataset, which is reflected in the KM curves. That is to say, the model doesn't pick up on the differences in HPV status since it was relatively not a huge contributor to the event outcomes.

5.3.3 Comparison With Other Studies

In [16], Wang et al. displayed KM curves for various models grouped by high and low risk. The risk score was made by averaging the scores of patients with and without event occurrence, to make high and low risk groups. These groupings are not directly comparable to the KM curves in this thesis. However, the KM curves in the study by Wang et al. showed a similar pattern to the KM curves in this thesis. The first time steps showed greater overlap of the curves and their confidence intervals than the later time steps. This is in accordance with the findings in this thesis, where the earlier time periods of the study was found to be harder to predict correctly.

Wang et al. [16] reported p-values from a log-rank test on the models' ability to differentiate between the KM curves. Notably, the model with the smallest $p - value$ was, for the OS endpoint, the PET+CT+GTV model. For the DM endpoint the model with the second smallest $p - value$ was the PET+CT+GTV model. These models are similar to the best overall performing model using the C-index in this thesis, which was the PET+CT+GTVp model. This could indicate the ability of multi-modal models to differentiate groups based on unseen clinical data.

In [51], Moan et al. reported KM curves grouped by low and high metabolic tumor volume. Two sets of KM curves were shown, one for HPV-related cancer and one without HPV-relation. The curves in Moan et al. are on the DFS endpoint, while the KM curves in this thesis use the OS endpoint, making the curves not directly comparable.

The KM curves in [51] are however made using the same data as the OUS dataset in this thesis. A similar trend where the curves are less distinguishable in the first time steps are found both in Moan et al. and this thesis. The KM curves for the HPV-unrelated patients show a clear separation, with a log-rank test $p - value$ of < 0.0001. Still, the curves are close to each other until around month 10, where they clearly separate. Even though the curves in the study by Moan et al. did not include the confidence intervals, they corroborates the findings made in this thesis where the model estimated KM curves were less separate in the first time intervals. This in turn justifies the choice of ten time intervals, focusing on the early stages of the follow-up period, which were more difficult to predict.

5.4 Assessment of Explainability Methods

5.4.1 The VarGrad method

In Figure 4.5, Figure 4.6 and Figure 4.7, as well as Figure 4.9, VarGrad heatmaps are seen overlaying the corresponding input channels. While the model was only provided with one input modality at a time, there were some interactions between the channels during training. For example, the CT+PET+GTVp model had its CT and PET masked images created by multiplying the original CT and PET with the GTVp tumor mask, making two new input images of the CT and PET images only within the masked area. This explains how the VarGrad heatmaps of the GTV contours could be found outside the delineated areas and exhibit patterns like the other input images. For example, the heatmap showing around the spinal area for the GTVp input channel in Figure 4.5, even though the GTVp input itself only includes information about the primary tumor and not the spinal area.

Interestingly, in the case where the PET image did not line up with the primary tumor and nodal areas in Figure 4.7, the VarGrad heatmap did not align with any nodal area, and while it did highlight the primary tumor, it was mostly found outside any delineation. This suggests that the model could rely mostly on the PET modality, since it was not able to use the CT image to find the nodal areas. The reason it was able to find the primary tumor could have been due to the GTVp contour being given as an input, and not from the CT image.

Modality correlation

Looking at the mean VarGrad plotted against the HU of the CT images CT, in Figure 4.8a, it is clear that the highest VarGrad values are seen over the highest HU values. This implies that the model may be associating certain density patterns on CT scans with survival outcomes, which could correlate with the tumor area. This density seems to be around 98HU and up, which corresponds to dense soft tissue [8].

The SUV plotted against the mean VarGrad values, in Figure 4.8b and Figure 4.10a, show a clear pattern of higher VarGrad values for higher SUV. VarGrad seems to be emphasizing areas with higher uptake, corresponding to areas with higher metabolic activity and therefore the cancer tumor area. This correlation was more pronounced than the correlation of VarGrad values against HU, seeing as the change in VarGrad values ranged from 0 to around 0.05 for HU, and from 0 to around 0.22 for SUV in the OS model and around 0.30 in the model for DFS. This VarGrad correlation difference shows that the model may prefer information from the PET modality over the CT modality, suggesting that PET is more important for survival diagnostic than CT.

Region correlation

The bar graph in Figure 4.8c shows in which region the VarGrad values were found, inside the primary tumor, nodal areas or outside either of them. The areas with VarGrad were considered important for the CT+PET+GTVp model predicting on the OS endpoint. A higher weight was placed upon the primary tumor area compared to the tumor nodes and other regions. This could reflect that survival prognosis on OS could be mostly determined by the primary tumor. On the other hand, for the PET-only model on DFS in Figure 4.10b, the primary tumor continued to be a focal point for the model, but there was a much higher emphasis on the tumor nodes, highlighting their potential role in disease progression and recurrence. The PET-only model also took more information from the *Other* regions, outside the tumor and nodal areas. The reason the model used this region might be that it was a model with no GTV contours. Without a GTV contour it might have been harder for the model to focus on specific regions. Also, the PET image by itself is of a lower resolution than the CT modality [10], which could lead to a more diffuse VarGrad heatmap.

Overall correlation

The violin plots in Figure 4.8d and Figure 4.10c provide a distribution of the correlation between VarGrad values and the mean SUV for all patients in the respective datasets. The height of the violin plots at various correlation levels indicates the density of patients with that particular correlation coefficient. For the CT+PET+GTVp model, the OUS correlation was wider than the MAASTRO, suggesting that the model's reliance on SUV to highlight areas of importance in the image varied more from patient to patient in the OUS dataset. For the PET-only model, the MAASTRO plot had a notably sharp peak, suggesting a strong agreement in correlation across this cohort. The tails of the violin plot represent the correlation coefficients that are less common within the dataset. The PET-only model for DFS shows both datasets as having short tails, suggesting that there were fewer patients with extremely high or low correlations. This could indicate that for DFS prognosis using a PET-only model, the relationship between the VarGrad highlighted regions and SUV was relatively stable across patients. The mean correlation coefficients were around 0.3 for both endpoints and datasets. This suggesting that, on average, SUV values of the PET modality might not be highly correlated with the areas the model assigns as important for prediction.

Comparison with other studies

A study by Huynh et al. [28], using radiomics for predicting OS and DFS outcomes, done on the same datasets as this thesis, showed similar patterns as this thesis in the VarGrad analysis done in the study. This thesis found a positive trend in mean VarGrad values with increasing SUV levels. The same trend was found in the study by Huynh et al., where both a CT+PET model on DFS and CT+PET+GTVp on OS showed this pattern. The relationship between VarGrad values and HU appears to be relatively low in both this thesis and the study. This indicates that PET images, more so than CT images, were used by the model for survival prognosis, both time-to-event and outcome prediction.

For VarGrad values distributed across primary tumor, nodes, and other regions, this thesis aligns with the study by Huynh et al. [28]. Both displayed the highest VarGrad values within tumor regions, affirming the model's prioritization of primary tumor characteristics over nodal regions or other factors. This reiterates the significance of primary tumor-based features in survival analysis. As for this thesis, Huynh et al. found the DFS model to have a relativity higher emphasis on the nodal regions. The consistency of the results from this thesis and the study by Huynh et al. reinforces the reliability of model interpretability. PET images emerged as a significant predictor of both DFS and OS.

Similarly to the VarGrad heatmap identified nodal regions as being more influential on predicting time until the DFS endpoint than the OS endpoint, the winner of the 2022 HECKTOR outcome prediction challenge [19], Rebaud et al., showed an interesting parallel. Rebaud et al. found that the radiomics feature number of affected nodal areas was impactful on predicting the recurrence free survival endpoint. While the endpoint recurrence free survival is different from the DFS endpoint, they are similar, both giving a measure of the time free from cancer after treatment. This suggests that the nodal regions could be more significant in predicting the time free from disease, more than the overall survival of the patient.

5.4.2 The SHAP method

Due to the limitation of the time-to-event models requiring 3D volumes as input, most SHAP methods were incompatible with the time-to-event model requirements, and only Gradient SHAP was found to be able to use 3D images as input. For example, the standard SHAP explainer, *shap.Explainer* from the SHAP library version 0.44.1 [49], required a masker function for the background data. The masker function for images, *shap.maskers.Image*, was incompatible with 3D volumes. Methods like Deep SHAP, Kernel SHAP and the standard SHAP explainer could have provided a different method for calculating feature importance, not based on perturbing parts of the image and looking at the model gradient. This would have provided a better range of methods when paired with the VarGrad method, giving a more nuanced model explainability assessment.

The Gradient SHAP method is similar to the VarGrad method in that it works by perturbing a sample and assigning importance to areas were perturbations resulted in changing predictions. Therefore, the SHAP plots are similar to the VarGrad plots. The thresholded Gradient SHAP values were the 1% most significant values for model prediction. There was a more noticeable spread out from the tumor and node areas in the thresholded SHAP values than the raw SHAP values.

Both the raw and thresholded Gradient SHAP values were shown mostly in and around the primary tumor and nodal areas in Figure 4.11, and in and around the primary tumor in Figure 4.12. Notably, the thresholded SHAP values seem to be more spread out, suggesting the most important areas for the model are found in and around the tumor and nodes.

Similarity between the outputs of the VarGrad method and the SHAP method could also indicate that the methods are correct in showing what features of the images are important to the model. Both methods highlighted the tumor and nodal areas, indicating that the model looks to those regions of the images for making time-to-event predictions.

5.5 Limitations

The model input requirements limited the number of usable explainability models. For example, Local Interpretable Model-agnostic Explanations (LIME) [66] was considered, but was not utilized as an explainability method due to its limitations. LIME required the data given to the explainer to be 2D images with three channels, like that of RGB images, while the models in this thesis require the input be 3D volumes with channels equal to the number of modalities and contours given to the model. Because of this conflict no LIME explainer could be used. This resulted in the usage of two similar explainability methods, VarGrad and Gradient SHAP, which did not cover a wide range of explainability methods.

Due to time constraints, not all model input combinations were tested. Not testing all possible combinations of CT, PET, GTVp and GTVn could have led to the wrong conclusions of modality and contour importance. Especially for the DFS endpoint, few model combinations were tested. No models predicting on the DFS endpoint contained the GTVn contour, even though the VarGrad method highlighted the nodal areas as being more important for the model predicting on the DFS endpoint than the OS endpoint. This limited the assessment of the GTVn contour as a model input.

Another limitation due to time constraints was the assessment of the SHAP explainability method. No statistical plots were made comparing SHAP values to the CT values, PET values or GTV delineation.

All models in this theses used a log likelihood loss function [37] [16]. While this approach is widely used, it comes with some limitations. By binning continuous time into intervals, there is an inherent loss of information, which could lead to a less accurate representation of the survival probabilities. The PET-only models ran on 10 and 20 time intervals were found to be robust against the choice of intervals, however, the choice of intervals could still impact the model's predictions. Having very large intervals could lead to a simplified model not able to detect nuances in the data [37]. On the other hand, short intervals may lead to overfitting by creating a model sensitive to noise.

Although the C-index is the most widely used metric for evaluating predictions in survival analysis [17] [18] [16], it falls short with high degrees of censoring [41]. As seen in Equation 2.17, uncertain ordered pairs are not counted, leading to an overly optimistic metric with high censoring proportions. Looking at Table 3.1, the OUS dataset had 61.4% of patients censored for the OS endpoint. With this high percentage of patients censored, the C-index may not necessarily be reliable, since the number of comparable pairs decreases when censoring increases. When this was the case IBS was consulted, which is a measure of accuracy of prediction, and therefore holds up under high degrees of censoring. In this thesis, the IBS scores gave around the same result as the C-index when comparing the models.

5.6 Future Work

Concerning the limitations of discrete time, an area for future work could be exploring continuous-time models, such as the Cox proportional hazards model [25] that can handle continuous time-to-event data. The inclusion of continuous time can give the model the ability to handle time-varying covariates [67]. Models with continuous time could also provide more accurate estimates of the hazard and survival probabilities [68].

Assessing different loss functions suited to time-to-event analysis is also worth pursuing. For example adapting a root mean square loss, common in regression models [69] [70], to a Brier-Score-based loss function that handles censored data, since the Time Dependent Brier Score Under Random Censorship essentially is an adaptation of root mean square taking censoring into account [44].

Another aspect of survival analysis not pursued in this thesis was the inclusion of left-censoring [21], where the beginning of the at-risk period is unknown. Models that incorporate left-censoring could be able to take in more information, ensuring that early events are not overlooked [71].

Chapter 6

Conclusion

This thesis aimed to develop time-to-event CNN models for predicting survival time until the OS and DFS endpoints. This was done for patients from the OUS hospital and the MAASTRO clinic. Seven models were made for predicting on the OS endpoint, and five were made for the DFS endpoint, using a variety of combinations of CT images, PET images, GTVp primary tumor contour and GTVn nodal area contour as input. The models were evaluated using the C-index, AUC and IBS metrics, over a five year period, splitting the follow-up time into ten time intervals.

An additional PET-only model was trained on a different split of the follow-up period into 20 intervals. This model achieved a C-index of 0.69, which was lower than the PET-only model trained on the 10 interval split, which had a C-index of 0.70. This suggested that the models performances were robust to the change of time intervals. Therefore, the five year split of 10 time intervals was chosen for all other models, focusing on the beginning of the follow-up period.

The CT+PET+GTVp model emerged as the highest performing model on the OS endpoint, with a C-index of 0.74 for the OUS dataset and 0.68 for the MAASTRO set. This model used anatomical data from the CT modality, metabolic information from the PET modality and the delineation of the primary tumor from the GTVp contour. This suggested that a multi-modal approach was most effective at capturing the nuances of the data and achieving a high performance on the OS endpoint.

When predicting on the DFS endpoint, the PET-only model had the highest C-index for both datasets, achieving a C-index score of 0.62 and 0.67 for the OUS and MAASTRO datasets, respectively. The predictions on the DFS endpoint were generally of a lower performance than predictions on OS. This suggested that DFS predictions may be influenced by more subtle and complex factors that are not as easily picked up by the models, making it harder to predict.

Models that utilized the PET modality consistently outperformed models without it, on both endpoints, suggesting that the models had a high reliance on the PET modality. The model with the lowest performance, across all metrics, was overall the CT-only model. This reinforced the notion that the PET modality provided more relevant information than the CT modality, for time-to-event predictions. The GTVp primary tumor contour was found to be a valuable model input, especially for predictions on the OS endpoint.

The second goal of this thesis was to asses the models on explainability, i.e. how the model predictions related to the observed data, and what parts of the images the models used for predictions. This was done using KM curves and saliency maps from the VarGrad and SHAP methods. The KM curves showed that the models generally overestimated the survival probability of all patients. However, the models significantly differentiated between the survival probabilities of different cancer stages and HPV status, whenever the differences were present in the observed data.

The VarGrad heatmaps showed a stronger correlation with the SUV from PET than the HU from CT, suggesting that the model relied more PET images than CT images. However, the distribution of the correlation coefficients between SUV and mean VarGrad values per patient showed that, while the PET modality was most correlated with VarGrad values, the correlation varies considerably from patient to patient. The mean correlation for all patients was around 0.3 for both endpoints and datasets, suggesting that, on average, SUV values of the PET modality might not be highly correlated with the areas the model assigned as important for prediction. The VarGrad highlighted areas were mostly within the primary tumor, and very little outside it, when predicting on the OS endpoint with the CT+PET+GTVp model. For the DFS endpoint, using the PET-only model, the VarGrad highlighted areas were almost equally found in the primary tumor and nodal regions, preferring the primary tumor, and some outside these areas. This suggested that the nodal areas were less important in prediction on the OS endpoint than the DFS endpoint. Due to the similar nature of the Gradient SHAP method to the VarGrad method, the SHAP heatmaps proved to follow a similar pattern to the VarGrad.

Bibliography

[1] Cancer Registry of Norway, "Cancer in norway 2023 - cancer incidence, mortality, survival and prevalence in norway.," 2024, Visited on 13.05.2024, ISSN: 0806-3621. [Online]. Available: https://www.kreftregisteret.no/globalassets/cancer-in-norway/2023/cin_report-2023.pdf.

[2] C. Yu, L. Li, S. Wang, *et al.*, "Advances in nanomaterials for the diagnosis and treatment of head and neck cancers: A review," *Bioactive Materials*, vol. 25, pp. 430–444, 2023, ISSN: 2452-199X. DOI: 10.1016/j.bioactmat.2022.08.010.

[3] A. Argiris, M. V. Karamouzis, D. Raben, and R. L. Ferris, "Head and neck cancer," *The Lancet*, vol. 371, no. 9625, pp. 1695–1709, 2008, ISSN: 0140-6736. DOI: 10.1016/S0140-6736(08)60728-X.

[4] A. Barsouk, J. S. Aluru, P. Rawla, K. Saginala, and A. Barsouk, "Epidemiology, risk factors, and prevention of head and neck squamous cell carcinoma," *Medical Sciences*, vol. 11, no. 2, 2023, ISSN: 2076-3271. DOI: 10.3390/medsci11020042.

[5] 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.

[6] M. Hashibe, P. Brennan, S.-c. Chuang, *et al.*, "Interaction between tobacco and alcohol use and the risk of head and neck cancer: Pooled analysis in the international head and neck cancer epidemiology consortium," *Cancer Epidemiology Biomarkers & Prevention*, vol. 18, no. 2, pp. 541–550, 2009. DOI: 10.1158/1055-9965.EPI-08-0347.

[7] E. Tumban, "A current update on human papillomavirus-associated head and neck cancers," *Viruses*, vol. 11, no. 10, 2019, ISSN: 1999-4915. DOI: 10.3390/v11100922.

[8] M. M. Ter-Pogossian, "Basic principles of computed axial tomography," *Seminars in Nuclear Medicine*, vol. 7, no. 2, pp. 109–127, 1977, ISSN: 0001-2998. DOI: 10.1016/S0001-2998(77)80013-5.

[9] D. Caramella, M. Revelli, and A. Villa, "Essentials of ct image interpretation," in *Nuclear Medicine Textbook: Methodology and Clinical Applications*, D. Volterrani, P. A. Erba, I. Carrió, H. W. Strauss, and G. Mariani, Eds. Springer International Publishing, 2019, pp. 281–316, ISBN: 978-3-319-95564-3. DOI: 10.1007/978-3-319-95564-3_14.

[10] S. Basu, T. C. Kwee, S. Surti, E. A. Akin, D. Yoo, and A. Alavi, "Fundamentals of pet and pet/ct imaging," *Annals of the New York Academy of Sciences*, vol. 1228, no. 1, pp. 1–18, 2011. DOI: 10.1111/j.1749-6632.2011.06077.x.

[11] J. P. Logue, C. L. Sharrock, R. A. Cowan, G. Read, J. Marrs, and D. Mott, "Clinical variability of target volume description in conformal radiotherapy planning," *International Journal of Radiation Oncology*Biology*Physics*, vol. 41, no. 4, pp. 929–932, 1998, ISSN: 0360-3016. DOI: 10.1016/S0360-3016(98)00148-5.

[12] J. A. Antolak and I. I. Rosen, "Planning target volumes for radiotherapy: How much margin is needed?" *International Journal of Radiation Oncology*Biology*Physics*, vol. 44, no. 5, pp. 1165–1170, 1999, ISSN: 0360-3016. DOI: 10.1016/S0360-3016(99)00117-0.

[13] L. Q. Chow, "Head and neck cancer," *New England Journal of Medicine*, vol. 382, no. 1, pp. 60–72, 2020. DOI: 10.1056/NEJMra1715715.

[14] R. J. Gillies, P. E. Kinahan, and H. Hricak, "Radiomics: Images are more than pictures, they are data," *Radiology*, vol. 278, no. 2, pp. 563–577, 2016. DOI: 10.1148/radiol.2015151169.

[15] V. Kumar, Y. Gu, S. Basu, *et al.*, "Radiomics: The process and the challenges," *Magnetic Resonance Imaging*, vol. 30, no. 9, pp. 1234–1248, 2012, ISSN: 0730-725X. DOI: 10.1016/j.mri.2012.06.010.

[16] Y. Wang, E. Lombardo, M. Avanzo, *et al.*, "Deep learning based time-to-event analysis with pet, ct and joint pet/ct for head and neck cancer prognosis," *Computer Methods and Programs in Biomedicine*, vol. 222, p. 106 948, 2022, ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2022.106948.

[17] V. Andrearczyk, V. Oreiller, M. Hatt, and A. Depeursinge, Eds., *Head and Neck Tumor Segmentation and Outcome Prediction, Third Challenge, HECKTOR 2022, Held in Conjunction with MICCAI 2022, Singapore, September 22, 2022, Proceedings*. Springer Cham, 2023, ISBN: 978-3-031-27420-6. DOI: 10.1007/978-3-031-27420-6.

[18] V. Andrearczyk, V. Oreiller, M. Hatt, and A. Depeursinge, Eds., *Head and Neck Tumor Segmentation and Outcome Prediction, Second Challenge, HECKTOR 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, Proceedings*. Springer Cham, 2022, ISBN: 978-3-030-98252-2. DOI: 10.1007/978-3-030-98253-9.

[19] L. Rebaud, T. Escobar, F. Khalid, K. Girum, and I. Buvat, "Simplicity is all you need: Out-of-the-box nnunet followed by binary-weighted radiomic model for segmentation and outcome prediction in head and neck pet/ct," in *Head and Neck Tumor Segmentation and Outcome Prediction*, V. Andrearczyk, V. Oreiller, M. Hatt, and A. Depeursinge, Eds., Cham: Springer Nature Switzerland, 2023, pp. 121–134, ISBN: 978-3-031-27420-6. DOI: 10.1007/978-3-031-27420-6_13.

[20] R. Flynn, "Survival analysis," *Journal of Clinical Nursing*, vol. 21, no. 19pt20, pp. 2789–2797, 2012. DOI: 10.1111/j.1365-2702.2011.04023.x.

[21] T. G. Clark, M. J. Bradburn, S. B. Love, and D. G. Altman, "Survival analysis part i: Basic concepts and first analyses," *British Journal of Cancer*, vol. 89, no. 2, pp. 232–238, 2003, ISSN: 1532-1827. DOI: 10.1038/sj.bjc.6601118.

[22] M. K. David G. Kleinbaum, *Survival Analysis, A Self-Learning Text, Third Edition*. Springer, 2011, ISBN: 978-1-4419-6645-2. DOI: 10.1007/978-1-4419-6646-9.

[23] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American Statistical Association*, vol. 53, no. 282, pp. 457–481, 1958, ISSN: 01621459. DOI: 10.1080/01621459.1958.10501452.

[24] N. Mantel, "Evaluation of survival data and two new rank order statistics arising in its consideration.," *Cancer chemotherapy reports*, vol. 50 3, pp. 163–70, 1966. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/5910392/.

[25] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 34, no. 2, pp. 187–202, 1972. DOI: 10.1111/j.2517-6161.1972.tb00899.x.

[26] P. Royston, B. Choodari-Oskooei, M. K. B. Parmar, and J. K. Rogers, "Combined test versus logrank/cox test in 50 randomised trials," *Trials*, vol. 20, no. 1, p. 172, 2019, ISSN: 1745-6215. DOI: 10.1186/s13063-019-3251-5.

[27] M. R. Salmanpour, S. M. Rezaeijo, M. Hosseinzadeh, and A. Rahmim, "Deep versus handcrafted tensor radiomics features: Prediction of survival in head and neck cancer using machine learning and fusion techniques," *Diagnostics*, vol. 13, no. 10, 2023, ISSN: 2075-4418. DOI: 10.3390/diagnostics13101696.

[28] B. N. Huynh, A. R. Groendahl, O. Tomic, *et al.*, "Head and neck cancer treatment outcome prediction: A comparison between machine learning with conventional radiomics features and deep learning radiomics," *Frontiers in Medicine*, vol. 10, 2023, ISSN: 2296-858X. DOI: 10.3389/fmed.2023.1217037.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org.

[30] H. Gu, Y. Wang, S. Hong, and G. Gui, "Blind channel identification aided generalized automatic modulation recognition based on deep learning," *IEEE Access*, vol. PP, pp. 1–1, 2019. DOI: 10.1109/ACCESS.2019.2934354.

[31] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, 2018, ISSN: 1869-4101. DOI: 10.1007/s13244-018-0639-9.

[32] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, Y. Lechevallier and G. Saporta, Eds., Heidelberg: Physica-Verlag HD, 2010, pp. 177–186, ISBN: 978-3-7908-2604-3. DOI: 10.1007/978-3-7908-2604-3_16.

[33] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. DOI: 10.48550/arXiv.1412.6980.

[34] Y. Zhang, E. M. Lobo-Mueller, P. Karanicolas, S. Gallinger, M. A. Haider, and F. Khalvati, "Cnn-based survival model for pancreatic ductal adenocarcinoma in medical imaging," *BMC Med Imaging*, vol. 20, no. 1, p. 11, 2020. DOI: 10.1186/s12880-020-0418-1.

[35] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, 2018, ISSN: 0360-0300. DOI: 10.1145/3236009.

[36] S. Azam, S. Montaha, K. U. Fahim, A. R. H. Rafid, M. S. H. Mukta, and M. Jonkman, "Using feature maps to unpack the cnn 'black box' theory with two medical datasets of different modality," *Intelligent Systems with Applications*, vol. 18, p. 200 233, 2023, ISSN: 2667-3053. DOI: 10.1016/j.iswa.2023.200233.

[37] M. F. Gensheimer and B. Narasimhan, "A scalable discrete-time survival model for neural networks," *PeerJ*, vol. 7, 2019. DOI: 10.7717/peerj.6257.

[38] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997, ISSN: 0031-3203. DOI: 10.1016/S0031-3203(96)00142-2.

[39] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006. DOI: 10.1016/j.patrec.2005.10.010.

[40] J. Harrell Frank E., R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati, "Evaluating the Yield of Medical Tests," *JAMA*, vol. 247, no. 18, pp. 2543–2546, 1982, ISSN: 0098-7484. DOI: 10.1001/jama.1982.03320430047030.

[41] H. Uno, T. Cai, M. J. Pencina, R. B. D'Agostino, and L. J. Wei, "On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data," *Statistics in Medicine*, vol. 30, no. 10, pp. 1105–1117, 2011. DOI: 10.1002/sim.4154.

[42] E. Longato, M. Vettoretti, and B. Di Camillo, "A practical perspective on the concordance index for the evaluation and selection of prognostic time-to-event models," *Journal of Biomedical Informatics*, vol. 108, p. 103 496, 2020. DOI: 10.1016/j.jbi.2020.103496.

[43] G. W. BRIER, "Verification of forecasts expressed in terms of probability," *Monthly Weather Review*, vol. 78, no. 1, pp. 1–3, 1950. DOI: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

[44] E. Graf, C. Schmoor, W. Sauerbrei, and M. Schumacher, "Assessment and comparison of prognostic classification schemes for survival data," *Statistics in Medicine*, vol. 18, no. 17-18, pp. 2529–2545, 1999. DOI: 10.1002/(SICI)1097-0258(19990915/30)18:17/18<2529::AID-SIM274>3.0.CO;2-5.

[45] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, "A benchmark for interpretability methods in deep neural networks," 2019. DOI: 10.48550/arXiv.1806.10758.

[46] M. Sundararajan, A. Taly, and Q. Yan, *Axiomatic attribution for deep networks*, 2017. DOI: 10.48550/arXiv.1703.01365.

[47] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2015. DOI: 10.48550/arXiv.1412.6806.

[48] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," 2020. DOI: 10.48550/arXiv.1810.03292.

[49] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017. DOI: 10.48550/arXiv.1705.07874.

[50] M. E. Charlson, P. Pompei, K. L. Ales, and C. MacKenzie, "A new method of classifying prognostic comorbidity in longitudinal studies: Development and validation," *Journal of Chronic Diseases*, vol. 40, no. 5, pp. 373–383, 1987, ISSN: 0021-9681. DOI: 10.1016/0021-9681(87)90171-8.

[51] J. M. Moan, C. D. Amdal, E. Malinen, J. G. Svestad, T. V. Bogsrud, and E. Dale, "The prognostic role of 18f-fluorodeoxyglucose pet in head and neck cancer depends on hpv status," *Radiotherapy and Oncology*, vol. 140, pp. 54–61, 2019, ISSN: 0167-8140. DOI: 10.1016/j.radonc.2019.05.019.

[52] A. R. Groendahl, I. S. 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.

[53] Y. M. Moe, A. R. Groendahl, O. Tomic, E. Dale, E. Malinen, and C. M. Futsaether, "Deep learning-based auto-delineation of gross tumour volumes and involved nodes in pet/ct images of head and neck cancer patients," *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 48, no. 9, pp. 2782–2792, 2021, ISSN: 1619-7089. DOI: 10.1007/s00259-020-05125-x.

[54] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," Proceedings of Machine Learning Research, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds., pp. 6105–6114, 2019. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html.

[55] V. Berisha, C. Krantsevich, P. R. Hahn, *et al.*, "Digital medicine and the curse of dimensionality," *npj Digital Medicine*, vol. 4, no. 1, p. 153, 2021, ISSN: 2398-6352. DOI: 10.1038/s41746-021-00521-5.

[56] Martín Abadi, Ashish Agarwal, Paul Barham, *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: https://www.tensorflow.org/.

[57] F. Chollet *et al.*, *Keras*, https://keras.io, 2015.

[58] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, *Understanding data augmentation for classification: When to warp?* 2016. DOI: 10.1109/DICTA.2016.7797091.

[59] A. Kleppe, O.-J. Skrede, S. De Raedt, K. Liestøl, D. J. Kerr, and H. E. Danielsen, "Designing deep learning studies in cancer diagnostics," *Nature Reviews Cancer*, vol. 21, no. 3, pp. 199–211, 2021, ISSN: 1474-1768. DOI: 10.1038/s41568-020-00327-9.

[60] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html.

[61] C. Davidson-Pilon, "Lifelines: Survival analysis in python," *Journal of Open Source Software*, vol. 4, no. 40, p. 1317, 2019. DOI: 10.21105/joss.01317.

[62] S. Pölsterl, "Scikit-survival: A library for time-to-event analysis built on top of scikit-learn," *Journal of Machine Learning Research*, vol. 21, no. 212, pp. 1–6, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-729.html.

[63] S. Sawyer, "The greenwood and exponential greenwood confidence intervals in survival analysis," *Applied survival analysis: regression modeling of time to event data*, pp. 1–14, 2003. [Online]. Available: https://www.math.wustl.edu/~sawyer/handouts/greenwood.pdf.

[64] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: Removing noise by adding noise," 2017. DOI: 10.48550/arXiv.1706.03825.

[65] N. Breslow and J. Crowley, "A Large Sample Study of the Life Table and Product Limit Estimates Under Random Censorship," *The Annals of Statistics*, vol. 2, no. 3, pp. 437–453, 1974. DOI: 10.1214/aos/1176342705.

[66] M. T. Ribeiro, S. Singh, and C. Guestrin, *"why should i trust you?": Explaining the predictions of any classifier*, 2016. DOI: 10.48550/arXiv.1602.04938.

[67] W. Sauerbrei, P. Royston, and M. Look, "A new proposal for multivariable modelling of time-varying effects in survival data based on fractional polynomial time-transformation," *Biometrical Journal*, vol. 49, no. 3, pp. 453–473, 2007. DOI: 10.1002/bimj.200610328.

[68] H. Kvamme and Ø. Borgan, "Continuous and discrete-time survival prediction with neural networks," *Lifetime Data Analysis*, vol. 27, no. 4, pp. 710–736, 2021, ISSN: 1572-9249. DOI: 10.1007/s10985-021-09532-6.

[69] R. Kumar, M. Gupta, P. Goplani, and Abhijit, "Analysis of invoice management system using regression techniques with improved loss functions," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2023, pp. 1–6. DOI: 10.1109/ICCCNT56998.2023.10308025.

[70] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)? – arguments against avoiding rmse in the literature," *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014. DOI: 10.5194/gmd-7-1247-2014.

[71] J. Karvanen, O. Saarela, and K. Kuulasmaa, "Nonparametric multiple imputation of left censored event times in analysis of follow-up data," *Journal of Data Science*, vol. 8, no. 1, pp. 151–172, 2022, ISSN: 1680-743X. DOI: 10.6339/JDS.2010.08(1).495.

Appendix A

Code Excerpts

A.1 Negative Log Likelihood Loss Class

```
@custom_loss
class NegativeLogLikelihood(Loss):
    def __init__(
            self, n_intervals, reduction="auto", name="negative_log_likelihood_loss"):
        super().__init__(reduction, name)
        self.n_intervals = n_intervals

    def call(self, target, prediction):
        # remove the last two elements, true time and true event
        target = target[:, :-2]
        # component for all individuals
        cens_uncens = 1. + target[:, 0:self.n_intervals] * (prediction - 1.)
        # component for only uncensored individuals
        uncens = 1. - target[:, self.n_intervals:2 * self.n_intervals] * prediction
        # return -log likelihood
        L = K.sum(-K.log(K.clip(K.concatenate((cens_uncens, uncens)),
            K.epsilon(), None)), axis=-1)
        return L
```

A.2 SurvArray Class

```
@custom_preprocessor
class MakeSurvArray(BasePreprocessor):
    def __init__(self, breaks):
            self.breaks = np.array(breaks)

    def transform(self, data, targets):
        t = targets[:, 1]
        f = targets[:, 0]
        n_samples = t.shape[0]
        n_intervals = len(self.breaks) - 1
        timegap = self.breaks[1:] - self.breaks[:-1]
        breaks_midpoint = self.breaks[:-1] + 0.5 * timegap
        y_train = np.zeros((n_samples, n_intervals * 2))
        for i in range(n_samples):
            # if failed (not censored)
            if f[i]:
                # give credit for surviving each time interval
                # where failure time >= upper limit
                y_train[i, 0:n_intervals] = 1.0 * (t[i] >= self.breaks[1:])
                # if failure time is greater than end of last time interval,
                # no time interval will have failure marked
                if t[i] < self.breaks[-1]:
                    # mark failure at first bin where survival time < upper break-point
                    y_train[i, n_intervals + np.where(t[i] < self.breaks[1:])[0][0]] = 1
            # if censored
            else:
                # if censored and lived more than half-way through interval,
                # give credit for surviving the interval.
                y_train[i, 0:n_intervals] = 1.0 * (t[i] >= breaks_midpoint)
                # add the true time and event data at the end
        return data, np.concatenate([y_train, targets], axis=-1)
```

A.3 AUC Class

```
class AUC_scorer:
    def __call__(self, y_true, y_pred, **kwargs):
        true = y_true[:, :10]
        return roc_auc_score(true, y_pred)
```

A.4 C-index Class

```
class HCI_scorer:
    def __call__(self, y_true, y_pred, num_year=5, **kwargs):
        event = y_true[:, -2]
        time = y_true[:, -1]
        no_time_interval = y_pred.shape[-1]
        breaks = np.arange(0, 61, 60//no_time_interval)
        predicted_score = np.cumprod(y_pred[:, 0: np.where(
            breaks >= num_year*12)[0][0]], axis=1)[:, -1]
        return concordance_index(time, predicted_score, event)
```

A.5 IBS Class

```
class IBS_scorer:
    def __call__(self, y_true, y_pred, **kwargs):
        event = y_true[:, -2]
        time = y_true[:, -1]
        survival_train = np.array(list(zip(event, time)))
        dtype = [('event', bool), ('time', np.float64)]
        structured_survival_train = np.array(
            list(map(tuple, survival_train)), dtype=dtype)
        times = [0, 6, 12, 18, 24, 30, 36, 42, 48, 54, 60]
        score = integrated_brier_score(structured_survival_train,
                                       structured_survival_train, y_pred, times)
        return score
```

Appendix B

Ensemble Model Performances

B.1 OUS Ensemble Model Performances

Table B.1: Ensemble model performances on the OUS dataset.

Test Fold	Endpoint	C-index	AUC	IBS
CT + PET				
4	OS	0.62	0.66	0.20
3	OS	0.66	0.63	0.18
2	OS	0.65	0.64	0.19
1	OS	0.84	0.76	0.12
0	OS	0.68	0.69	0.20
PET				
4	OS	0.61	0.63	0.21
3	OS	0.67	0.65	0.18
2	OS	0.76	0.67	0.17
1	OS	0.77	0.73	0.13
0	OS	0.72	0.67	0.18
PET + 20 intervals				
4	OS	0.65		
3	OS	0.67		
2	OS	0.73		
1	OS	0.78		
0	OS	0.68		
PET + GTVp				
4	OS	0.70	0.70	0.19
3	OS	0.80	0.72	0.15
2	OS	0.79	0.71	0.17
1	OS	0.72	0.73	0.14
0	OS	0.71	0.69	0.17
PET + GTVp + GTVn				
4	OS	0.69	0.73	0.20
3	OS	0.75	0.73	0.14
2	OS	0.87	0.80	0.15
1	OS	0.78	0.66	0.13
0	OS	0.63	0.68	0.21
CT				
4	OS	0.61	0.51	0.19
3	OS	0.68	0.65	0.18
2	OS	0.69	0.65	0.19
1	OS	0.75	0.68	0.16
0	OS	0.69	0.69	0.20
CT + PET + GTVp				

Continued on next page

Table B.1 – *Continued from previous page*

Test Fold	Endpoint	C-index	AUC	IBS
4	OS	0.73	0.78	0.18
3	OS	0.75	0.66	0.16
2	OS	0.81	0.78	0.16
1	OS	0.75	0.73	0.13
0	OS	0.75	0.65	0.19
CT + PET + GTVp + GTVn				
4	OS	0.59	0.60	0.20
3	OS	0.76	0.71	0.15
2	OS	0.65	0.64	0.17
1	OS	0.68	0.68	0.16
0	OS	0.78	0.73	0.19
CT + PET + GTVp				
4	DFS	0.54	0.46	0.26
3	DFS	0.61	0.63	0.23
2	DFS	0.79	0.61	0.20
1	DFS	0.47	0.49	0.26
0	DFS	0.68	0.63	0.23
CT + PET				
4	DFS	0.53	0.52	0.25
3	DFS	0.52	0.60	0.25
2	DFS	0.72	0.56	0.21
1	DFS	0.60	0.57	0.23
0	DFS	0.73	0.57	0.23
PET				
4	DFS	0.69	0.62	0.25
3	DFS	0.51	0.51	0.26
2	DFS	0.77	0.59	0.19
1	DFS	0.55	0.52	0.25
0	DFS	0.74	0.66	0.21
PET + GTVp				
4	DFS	0.70	0.61	0.25
3	DFS	0.61	0.60	0.22
2	DFS	0.79	0.60	0.21
1	DFS	0.42	0.47	0.30
0	DFS	0.66	0.65	0.21
CT				
4	DFS	0.48	0.46	0.25
3	DFS	0.55	0.58	0.24
2	DFS	0.60	0.51	0.23
1	DFS	0.57	0.56	0.24
0	DFS	0.62	0.58	0.24

B.2 MAASTRO Ensemble Model Performances

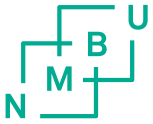
Table B.2: Ensemble model performances on the MAASTRO dataset.

Test Fold	Endpoint	C-index	AUC	IBS
CT + PET				
4	OS	0.64	0.58	0.18
3	OS	0.66	0.64	0.17
2	OS	0.62	0.60	0.17
1	OS	0.61	0.57	0.18
0	OS	0.63	0.60	0.18
PET				
4	OS	0.67	0.64	0.17
3	OS	0.66	0.65	0.17
2	OS	0.70	0.66	0.18
1	OS	0.67	0.63	0.17
0	OS	0.65	0.61	0.17
PET+GTVp				
4	OS	0.66	0.60	0.17
3	OS	0.64	0.63	0.17
2	OS	0.64	0.64	0.18
1	OS	0.65	0.65	0.19
0	OS	0.66	0.65	0.17
PET+GTVp+GTVn				
4	OS	0.65	0.63	0.17
3	OS	0.67	0.66	0.17
2	OS	0.67	0.64	0.17
1	OS	0.63	0.60	0.18
0	OS	0.64	0.63	0.17
CT				
4	OS	0.62	0.60	0.18
3	OS	0.61	0.58	0.18
2	OS	0.62	0.59	0.18
1	OS	0.61	0.57	0.19
0	OS	0.62	0.57	0.18
CT+PET+GTVp				
4	OS	0.68	0.66	0.17
3	OS	0.66	0.65	0.17
2	OS	0.66	0.66	0.17
1	OS	0.68	0.69	0.17
0	OS	0.67	0.63	0.17
CT+PET+GTVp+GTVn				
4	OS	0.67	0.63	0.17
3	OS	0.69	0.66	0.16
2	OS	0.71	0.70	0.16
1	OS	0.67	0.65	0.18
0	OS	0.66	0.64	0.17
CT+PET+GTVp				
4	DFS	0.59	0.57	0.22

Continued on next page

Table B.2 – *Continued from previous page*

Test Fold	Endpoint	C-index	AUC	IBS
3	DFS	0.64	0.61	0.21
2	DFS	0.63	0.62	0.21
1	DFS	0.62	0.63	0.21
0	DFS	0.62	0.61	0.22
CT+PET				
4	DFS	0.63	0.62	0.21
3	DFS	0.66	0.66	0.21
2	DFS	0.63	0.62	0.22
1	DFS	0.63	0.64	0.21
0	DFS	0.62	0.59	0.22
PET				
4	DFS	0.67	0.62	0.21
3	DFS	0.66	0.62	0.21
2	DFS	0.65	0.61	0.22
1	DFS	0.65	0.61	0.22
0	DFS	0.64	0.61	0.22
PET+GTVp				
4	DFS	0.59	0.60	0.22
3	DFS	0.62	0.62	0.22
2	DFS	0.60	0.60	0.22
1	DFS	0.61	0.62	0.23
0	DFS	0.61	0.60	0.22
CT				
4	DFS	0.65	0.60	0.21
3	DFS	0.64	0.60	0.21
2	DFS	0.63	0.58	0.22
1	DFS	0.63	0.61	0.22
0	DFS	0.60	0.58	0.22



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

Postboks 5003
NO-1432 Ås
Norway