university of groningen

faculty of science and engineering

Autoencoders for Head and Neck Cancer Prognosis

BACHELOR THESIS

Author:

K.L.S Moerman

Primary supervisor:

Prof. Estefanía T. Martínez

Secondary supervisor:

Prof. George Azzopardi

Abstract

Radiation therapy is often used to treat head and neck cancer. Although locoregional treatments are quite efficient, there exists a possibility for the development of distant metastases or second primary cancers. It is therefore of significance to develop a model that can predict the recurrence of such a cancer. With said technology, it would become possible to propose a better suited treatment in relation to the cancer's current stage. We propose to devise a deep neural network in order to enhance the performance of traditional radiomics in the risk assessment of locoregional recurrence (LR), distant metastases (DM), and overall survival (OS) in head and neck cancer patients.

Accordingly, in order to assess the role of deep learning in this field we will conceive a structured scientific comparison between distinct network architectures. Notably, the classification prediction of radiomic and clinical attributes are based on a simple convolutional autoencoder and a denoising autoencoder. These will comply on identical structures in order to have an in depth analysis on the resulting performance.

Independent validation of the prediction and prognostic performance from a series of experiments was carried out. Indeed, a supplementary denoising criterion allowed the model to extract more representative features for the classification of the cancer prognosis (LR : 0.64, DM: 0.67, and OS: 0.69). Such methods could have a significant clinical impact by assisting health care professionals with choosing the correct personalised therapy.

Contents

List of Figures

List of Tables

1 Introduction

In the past years, machine learning has made a significant impact in most aspects of the Science, Technology, Engineering, and Mathematics fields. Deep learning, a unique subfield of machine learning, presents a new concept on learning representations from data by emphasising the use of successive layers. With regards to image processing, these layered representations are typically learned via models called Convolutional Neural Networks (CNN). Here, the term neural network is a reference to a simplification of the human brain's structure and function. These models have achieved noticeable progress in many computer vision applications by analysing, processing, and understanding images. Medical image processing, being of particular interest, benefit from such methods by extracting abstract features from high-dimensional data.

The survival rate of a diagnosed cancer patient is mostly associated with the cancer type and its current stage of development. Despite advanced treatment options, it is estimated that half of patients die from the cancer or the applied treatment [24]. Those that survive cancer have an increased risk to develop a second primary cancer. This is believed to be the result of the probability for a new cancer to develop, the possibility of the cancer surviving the treatment, or the same risk factors that were at the origin of the first cancer [1]. Predicting the likely development of a cancer or the survival of a patient, also called prognosis, depends on many factors. The most notable are the cancer type and the patient's overall health. However, this data seem to not be sufficient in order to predict cancer recurrence with certainty. As will be discussed in the next section, to determine an accurate prognosis it becomes more reliable to adopt proficient computational methods.

Being at the heart of this problem, neural network models can learn meaningful representations of input data, in this case medical images, to accurately predict or estimate the presence of cancer bio-markers. This paper explores such applications with radiomics. The term of ”radiomics” was first mentioned by Lambin et al. [11] to describe the automatic segmentation, feature extraction, and analysis in cancer imaging data. With the help of machine learning techniques, it is now possible to highlight specific patterns that may not be visible to the human eye. As a result, the objective for this procedure is to provide a clinical decision support system, meaning that a proficient computerised system can assist health care professionals with choosing the correct personalised therapy and, hopefully, obtain more accurate prognosis of the disease [2].

It is also worth mentioning that the relationship between processed images using radiomics and external clinical data is a key aspect in machine learning to improve medical treatment. Such techniques are being used to greater extent due to two main factors; first, as seen by Moore's Law, modern computers have increasing processing power since the introduction of parallel processing. The second becomes apparent by the large scale public databases that provide an open-source platform for researchers to analyse clinical data and share their results.

One model that is of particular interest for feature extraction on medical images are autoencoders. They are an unsupervised machine learning method as they do not need explicit labels to train. On the contrary, in the process to reconstruct the original input, they generate their own labels from the training data while preserving the most relevant aspects of the data. This model type has been popular for many years with the first application in the 1980s by Hinton and the PDP group to overview “back-propagation without a teacher” [8]. Nowadays, however, they are used in a wider range of applications. These include dimensionality reduction, feature learning, but more generally the ability to learn generative models of data. Their wide range of capability make them widely appealing to even some of the most powerful deep neural networks in the 2010s [4].

To advance towards more precise treatment options in radiotherapy, medical imaging together with clinical parameters are used to predict treatment failures in cancer patients. This paper proposes to develop an image-based model for treatment outcome prediction and risk factor discovery. Based on the recent research in the field we believe the most suitable neural network for this task are autoencoders as they reduce data dimensions by learning how to ignore noise. Accordingly, using such a model could potentially allow us to find a cause-and-effect relationship between a biological factor and cancer, but more importantly give a model based on our data instead of predefined filters [14]. This leads to our main research problem; we wish to determine whether autoencoders can help with the classification in cancer prognosis and as a result predict the tumour recurrence after treatment with increased accuracy.

After carefully implementing this model, we expect to contribute to the knowledge of dimensionality reduction with medical imaging for cancer recurrence prediction. Accordingly, in order to assess the performance of our autoencoders for this type of data, we will contrast our findings with the results of other related studies. On a broader scale, we anticipate that the prognosis outcome following a treatment is linked to distinct traits found within the cancerous tumour. These can then be visualised by reducing high-dimensional data to a latent space. By means of multivariate analysis, such as principal component analysis (PCA), potential features can be extracted in an informative approach. As such, it becomes possible to build an accurate and reliable model that can be used in the medical field to recommend additional personalised treatment options.

This study will approach the previously mentioned problem in a clear and structured report. To have a better overview on the final results, we start by exploring the current state of the art analytical methods in cancer prognosis survival using neural networks. This leads to a in depth description of the used methods that are applied to the data at hand. Then, the decisions taken that lead to our results will be explained in the experiments section; these take into consideration the description of the data, the metrics used to assess the model, and different networks that were compared. We conclude this paper by discussing the achieved results along with their implication in the medical field.

2 Related Works

Cancer is the leading cause of death worldwide [24]. Today, Computed Tomography (CT) based image features are commonly used to predict treatment failures on cancer patients. This is of significance as, according to the department of radiation oncology at the University Medical Centre of Groningen (UMCG), 30-50% of the patients with a locally advanced cancer stage still experience treatment failures. Thus, an outcome prediction is vital for the optimisation of treatment and can eventually lead to personalised treatment options.

While chemotherapy exposes the entire body to cancer-fighting medication, radiation therapy follows a more local treatment, focused at the part of the body affected by the tumour. However, such procedures do not take into consideration specific traits of the cancer [21]. To obtain a more precise outcome from the treatment we can use various computational methods. In the domain of medicine, methods such as radiomics are already in use to discover cancer features that are undetectable to the naked eye. With the help of data-characterisation algorithms, radiomics can extract large amounts of data from radiographic medical images [19]. As such, by using this method it is possible to detect hidden parts of the cancer and have a more focused procedure. Yet, recently more research is leaning towards the use of CNNs to enhance the detection of cancer features. Such model is made up of multiple layers that extract progressively more complex features from the data to produce an abstract but informative classification of the input [20]. Although these are promising to analyse clinical data, there is a lack of research on the approach to incorporate distinct patient data while using the optimal design to encode this one. It is therefore of interest to employ distinct model architectures on an established dataset. This paper will focus on the treatment outcome prediction of head and neck cancer based on features extracted by a convolutional autoencoder.

Current state-of-the-art deep learning methods support the progress in many medical areas; for instance, the classification of skin cancer type [12], the identification of cancer in histopathological slides [9], or even the detection of cancer bio-markers using nuclear morphometric measures [18]. Such developments are also extensively present in cancer prognosis studies where CNNs are used to classify cancerous cells for survival prediction. Glioblastoma is the most aggressive type of cancer that originates within the brain. Still, it appears research has found the O^6-methylguanine DNA methyltransferase (MGMT) gene to have a direct correlation with survival and is highly responsive to specific treatment options. However, any invasive procedure in the brain is not an easy task and must be carried out with caution. In an attempt to simplify this task, Korfiatis et al. [10] used a pre-trained 50 layer Residual Network (ResNet) with MRI images to predict the status of the MGMT gene, determined to be a predictor for brain tumour prognosis. Compared to other similar architectures, this model was the most efficient with an accuracy close to 95%. Similarly, another research built a bidirectional convolutional recurrent neural network to anticipate patients sensitivity to the planned treatment [6]. They achieved a high training accuracy of 97%, however both the validation and the test accuracy were only at 67% and 62% respectively despite various attempts to reduce over-fitting to the training data.

Other deep learning methods that have been applied to various cancer types have also shown promising potential in cancer prognosis. As medical data has different formats – including genomic data, clinical data, and radiomic data – it is important to take such information into account when defining a model architecture. Demonstrated by the above-mentioned research, when trying to encode a given unexplored dataset, a neural network that has already been trained and evaluated on a larger dataset is used as a base. This *transfer learning* is then fine-tuned to adhere to the concerned input [5].

Although this approach is promising, similar results ca be achieved by carefully training a CNN to recognise radiomic features. This is the case for a study that aimed to use a single end-to-end CNN to predict prognosis for head and neck squamous cell carcinoma (HNSCC) [3]. A considerable advantage to training such a model is the possibility to learn abstract features unique to the dataset at hand without the help from any secondary machine learning algorithms. Similarly, Zhaoa et al. have conceived an autoencoder to predict the disease progression of HNSCC patients [27]. RNA sequencing, miRNA sequencing, and methylation data were used as input instead of CT-images. As such, they were able to find a collection of features that are involved in the development of cancer. This model would assist in the improvement of prognosis and, as a result, be able to help in the proposal of varying treatment procedures for HNSCC patients depending on the cancer's current stage. A high-risk patient would need a more aggressive treatment compared to a low-risk patient, needing less radiation to minimise any side effects.

In order to have a representative overview on the problem, this paper aims to compare the achieved results to related research performed on the same data. Vallières et al. [25] have devised a prediction model using random forest and imbalance adjustment strategies to evaluate the risk of locoregional (LR) recurrence and distant metastasis (DM) before applying the suggested treatment. To achieve proficient prediction assessment, this machine learning approach combines both radiomics

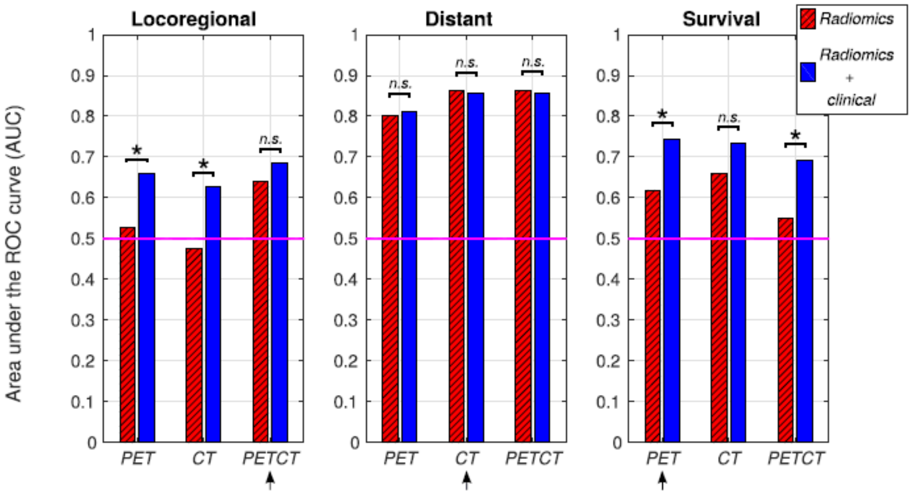


Figure 1: Prediction performance by Vallières et al.

The predictive results for three distinct radiomic feature sets: I- Positron Emission Tomography radiomic features (PET); II- CT radiomic features (CT); III- PET and CT radiomic features (PETCT). The outcome is determined by the Area Under the Curve metric. More on this in the *Experiments* section.

features – including the intensity, shape, and texture of the tumour – and clinical attributes – such as the patient's age, head & neck type, and tumour stage. Indeed, as the results from Figure 1 suggest, a more accurate risk indicator is realised when including radiomic features to a clinical prediction model. This leads us to believe that the combination of quantitative features in medical images with distinct categorical data in machine learning have a significant impact on the characteristics by which cancer recurrence can be identified, thereby allowing a more personalised treatment plan for patients.

With an overview of diverse machine learning approaches on the application of cancer prognosis, this paper wants to determine if another model, such as autoencoders, would be more efficient with the given data. The aim of an autoencoder is to create a reduced representation for a set of data using unsupervised networks. This model works together with a reconstruction side, where the autoencoder tries to reproduce the original input from the simplified encoding [15]. Various successful architectures of autoencoders have recently surfaced in order to generate a simple latent space such as Sparse or even Denoising autoencoders. Autoencoders often encode and decode the data through a single layer, but using a deeper architecture has many advantages. Experimentally, deep autoencoders produce a better abstraction of the data compared to shallow or linear autoencoders [7]. Similar radiomic analysis on medical grey-scale images have been performed to which it is shown that these filters are in fact able to effectively process and distinguish radiomic features without an explicit definition of any feature [3]. Furthermore, autoencoders have proven to be reliable and robust in predicting the outcome for HNSCC treatment using multi-omics data [27]. Therefore, it is of interest to determine the performance of autoencoders with the provided medical images.

Disregarding the achieved results of this research, by manipulating the data in a particular manner will give insight on the topic and will thus help to eventually develop a competent model that can improve the success rate of cancer treatments by following a more localised therapy.

3 Proposed Methodology

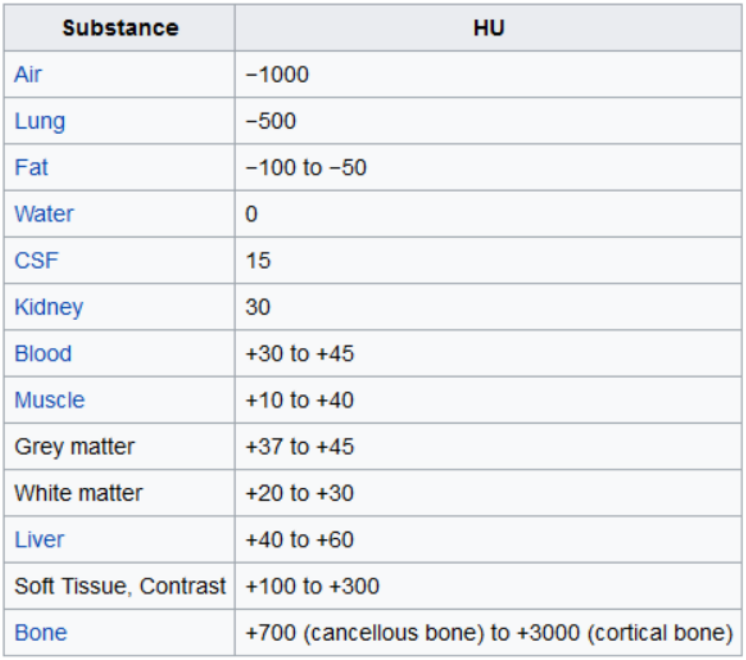
The proposed autoencoder model is presented in this section. First, a general overview of the data with the performed transformations is given. This is followed by a detailed description of the model with explanations for the design choices.

3.1 Data Structure

The data collection contains clinical data and computed tomography (CT) imaging data from a variety of institutions. A group of experienced radiation oncologist manually delineated the gross tumour volume on the scan of the concerned patients. This data is stored as object types in publicly available archives as to provide repeatability and reproducibility in research. These will be discussed in great detail below.

3.1.1 Digital Imaging and Communications in Medicine

As discussed prior, this paper focuses on the analysis of medical images through the processing of a convolutional autoencoder. The Digital Imaging and Communications in Medicine (DICOM) is today widely accepted and used in radiology for diagnostic imaging. It has been extended for various sub-specialities, including radiation therapy, and is known as DICOM-RT. It is important to understand how such data is constructed. In these files, there is a difference between what is represented and what is actually stored at the level of the pixels. The actual pixel of each slice in a CT volume are the Hounsfield Units (HU) of the slice. The scale of Figure 2 describes the absorption rate of x-rays when the CT-scan was performed. It is then possible to look at a slice through a DICOM viewer. This represents the pixel as grey values.



Substance	HU
Air	−1000
Lung	−500
Fat	−100 to −50
Water	0
CSF	15
Kidney	30
Blood	+30 to +45
Muscle	+10 to +40
Grey matter	+37 to +45
White matter	+20 to +30
Liver	+40 to +60
Soft Tissue, Contrast	+100 to +300
Bone	+700 (cancellous bone) to +3000 (cortical bone)

Figure 2: Hounsfield Unit scale

There are significant varieties in the types of information required for radiation therapy, and thus different categorisation; these include RT Structure Set, RT Plan, RT Dose, RT Image, and RT Treatment Record. However, since we are mostly interested in image data, only the RT Structure Set and RT Image objects will be discussed. A DICOM data object is made of several common attributes – including items such as ID, modality or even patient information – and the image pixel data. This latter attribute is unique, and thus for many modalities corresponds to an array of ”frames” to be able to represent data in three dimensions.

The RT Structure Set is an object that defines a collection of important delineated areas for radiation therapy, also known as Region of Interest (ROI). Such instances are body contours, tumour volumes (e.g. gross target volume, clinical target volume, planning target volume, etc.), organs at risk, and many others. The most relevant ROI to our study is the Gross Tumour Volume (GTV), this includes the entire physical tumour that is present on physical examination with endoscopy and imaging. Each structure will be associated with a frame of reference, with or without reference to the diagnostic images.

In contrast to a DICOM image object, RT Image includes not only image information but also the description of the image; meaning the position, plane, and orientation of the image. If required, the RT Image can also include the table position, isocentre position, patient position, and the type of device used to limit the radiation therapy beam. However, most of this information is not necessary to train and validate a deep learning model.

3.1.2 Quebec Institutions

This collection contains FDG-PET/CT and radiotherapy planning CT images of 298 patients with HNSCC from four different institutions in Quebec. The dates of the data, ranging from April 2006 to November 2014, have been modified as to retain anonymity of the patients. In the original study, cases with recurrent head and neck cancer or exposing metastases at presentation were not included in the dataset. From the patients involved, some 16% received radiation treatment while 84% received chemo-radiation with curative intent. The average follow-up period for the patients was 43 months. Patients that had a follow-up smaller than 24 months and that did not develop a locoregional recurrence or distant metastases during that period were excluded from the study. During the follow-up period, 15% of the patients developed a locoregional recurrence, 13% developed distant metastases and 19% of the patients died. A small individual overview is provided below, for a more complete list see Table 3 in the *Appendix*. Note, any retrospective analysis was carried out with respect to the guidelines and regulations as approved by the Research Ethics Committee of McGill University Health Centre (Protocol Number: MM-JGH-CR15-50).

- The Hôpital Général Juif (HGJ) de Montréal cohort is composed of 92 patients with primary HNSCC. These patients were treated by radiation or chemo-radiation therapy. Before the next verification, 12 patients developed locoregional recurrence (13%), 16 patients developed distant metastases (17%) and 14 patients died (15%). This data was used in the training set of the model.

- The Centre Hospitalier Universitaire de Sherbrooke (CHUS) cohort is composed of 102 patients with primary HNSCC. These patients were treated by radiation or chemo-radiation therapy. Before the next verification, 17 patients developed locoregional recurrence (17%), 10 patients developed distant metastases (10%) and 18 patients died (18%). This data was used in the training set of the model.

- The Hôpital Maisonneuve-Rosemont (HMR) de Montréal cohort is composed of 41 patients with primary HNSCC. These patients were treated by radiation or chemo-radiation therapy. Before the next verification, 9 patients developed locoregional recurrence (22%), 11 patients developed distant metastases (27%) and 19 patients died (46%). This data was used in the testing set of the model.

- The Centre Hospitalier de l'Université de Montréal (CHUM) cohort is composed of 65 patients with primary HNSCC. These patients were treated by radiation or chemo-radiation therapy. Before the next verification, 7 patients developed locoregional recurrence (11%), 3 patients developed distant metastases (5%) and 5 patients died (8%). This data was used in the testing set of the model.

3.1.3 Maastricht Institution

This collection contains radiotherapy planning CT images of 136 patients with HNSCC from the Maastricht University Medical Centre (MAASTRO). During the follow-up period, 18% developed a locoregional recurrence, 6% developed distant metastases and 54% of patients survived. For a more complete list refer to Table 4. As an attempt to obtain a generalised overview on this research, the data will be used as a validation set to determine the true efficiency of the trained model. Before performing transformations on the data, the same pre-processing methods discussed below will be applied to both public collections.

To have an accurate comparison with separate studies that use this dataset, a NumPy random seed was determined to have consistent training and testing splits. Essentially it provides a pseudo-random number generated by the computer. Future studies can use the following seed: `1234`.

3.1.4 Preprocessing

The data is organised by patient; they include the RT Structure Set and corresponding RT Image objects. It is important to note that such DICOM objects contain metadata with information about the image; this includes the size, dimensions, bit depth, modality used to create the data, and equipment settings used to capture the image. This is what is referred to as the file's meta information. With the help of the `pydicom` library, the metadata is easily accessible through function calls. They return a collection of $\{key : value\}$ pairs, where the key is the DICOM tag and the value is a DataElement instance.

The DataElement class is a simple object which keeps track of the following:

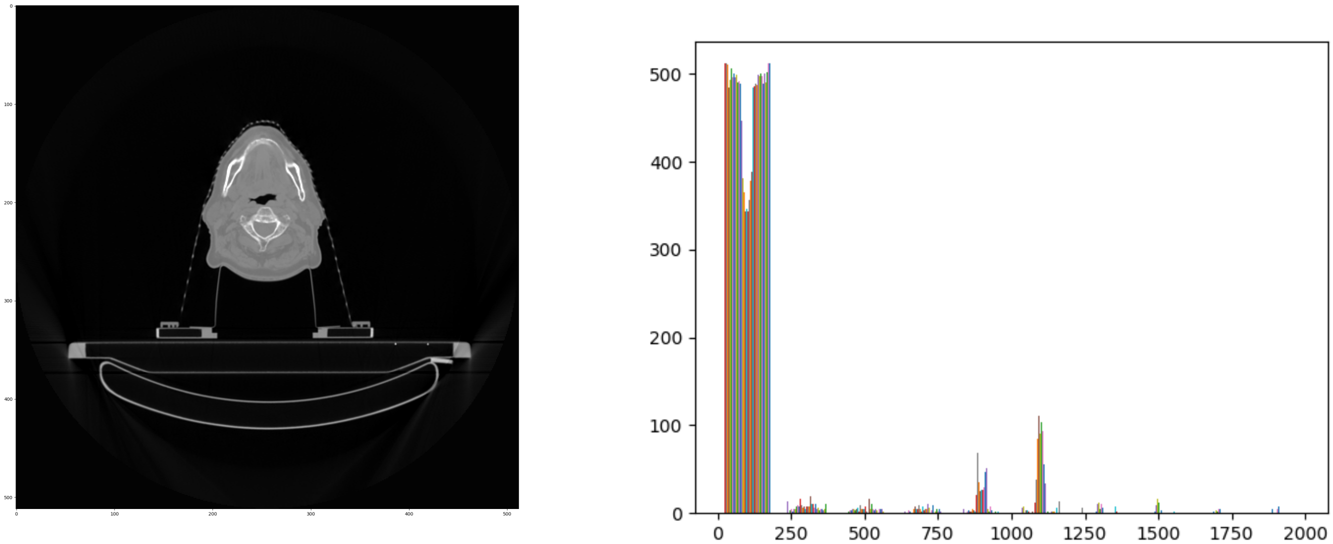
- **tag** – a DICOM tag; usually in the format (XXXX,XXXX) with hexadecimal numbers.

- **VR** – value representation; data type and format of that data element's values.

- **VM** – value multiplicity; specifies the number of values that can be encoded in the `value` field of that data element.
- **value** – the actual data; this is usually a sequence of strings or numbers.

It is recommended to access data elements using DICOM keywords, but as mentioned it is also possible to use the tag number directly. For simplicity we will use the former. As such, it allows to access the DICOM sequence of specific objects. The items inside the sequence are referenced by an index, similar to python's list data type. The only additional functionality is to make string representations more elegant. In order to work with the pixel data of an item, the raw bytes are available through the `PixelData` tag. Other useful tags to convert the data sequence to a list of pixels are the `PixelSpacing` and `SliceThickness` tags. Together with the dimensions of the image, these attributes allow to save the metadata of interest as a NumPy array of pixel intensities.

This results in a three dimensional data format; the number of pixel rows in a slice by the number of pixel columns in a slice by the number of slices. Different approaches need to be taken depending on the structure of the data. For this research, we will only be working on the prognostic outcome prediction based on 2D files. When comparing the result with the models applied on 3D files, it can be of interest to determine possible differences in performance. Analysis on such format is covered in the study undertaken by other students at the University of Groningen, notably Igor Pidik [17] and Maria-Sophia Stefan [23]. Both methods have their advantages; namely when using 3D images the deep learning model has a more complete overview on the features corresponding to the tumour, whereas when using 2D images there is greater flexibility to establish a diverse input. Accordingly, with an average of 28 slices per gross tumour volume in the Quebec collection, it is



(a) Head and Neck 2D image prior to modifications

(b) Head and Neck pixel intensity histogram

Figure 3: Original Head and Neck image with the corresponding pixel intensities

possible to select a variety of scattered images so that they appear as additional unique patient. Let us illustrate this concept by taking only 1 slice per patient. This will lead the model to train with a mere 190 CT-images. On the other hand, with a total of 4 disconnected slices per patient, the training set increases to a notable 733 CT-images. In general, the training data quantity required increases depending on the complexity of the problem at hand. With a small high dimensional input set, the model tends to over-fit; meaning that the model memorised specific patterns specific for the training data which in turns renders the learning process incompetent.

Now that the training images are defined it is also important to appropriately format the data that will be fed to the model. The CT-images are 512 pixels wide by 512 pixels high. However, as illustrated by Figure 3a, most of these pixels are responsible to display air and are therefore insignificant; i.e. have a pixel intensity of zero. The deep learning model will have to learn this characteristic by gradient descent, despite them not being relevant to the classification of cancer prognosis. For this reason, they are resized to 128 by 128 pixels and centred around the tumour. As discussed previously, each pixel present in such tensor takes a wide range of values to represent X-Ray beam intensities. Since this scale is heterogeneous, the data should either be normalised or standardised. Linear parameters, found in the RT Image header, are used for scaling the raw pixel values to HU, or vise-versa. Having intensities for corresponding tissues match between image sets can also have an impact on applications requiring image registration when methods of image similarity calculation used during image registration rely directly on pixel intensity values. For the DICOM format, the conversion parameters are the slope, s, and the intercept, i. A linear conversion is then applied to transform the raw pixel value to its HU value:

$$H(x) = s \cdot x + i$$

This study will make use of the raw pixel values of the CT-images, usually within the [0,4096] range. Thus, to properly normalise the tensors, all raw pixel values that exceed the bone radiodensity are restricted to a ceiling value; i.e. 2000. As a result, the input data is in the [0, 1] range and can be fed to the neural network.

The splitting between training and testing set is determined in accordance with the study performed by Vallières et al. [25]. They meticulously split the four Quebec institutions into two similar groups; one responsible for the model training (HGJ and CHUS; n=190), while the other is used for testing purposes (HMR and CHUM; n=103). By doing so, it allowed to:

- Generalise the training of the model to a variety of institutions.
- Create a ratio of approximately 2:1 between the training set and the testing set.
- Approximately maintain a proportional sampling of occurrence of events in the training and testing sets.

Following, in order to accurately determine the models capability, the MAASTRO dataset will be used as validation to take into account some institutional variability. Accordingly, this insight illustrates the potential scope of application that can be brought to the medical field.

3.2 Model architecture

Up to this point the data has been transformed into tensors with acceptable training and testing splits. It is now pertinent to expand on the methods used to build a model that will handle the discussed input.

3.2.1 Tools

Machine learning is a very complex field of study that requires many years of practice. With the help of machine learning frameworks, such as TensorFlow, the process of acquiring data, training models, and obtaining predictions becomes far more accessible. TensorFlow is an open source library for numerical computation used by neural networks. It uses Python for building a convenient front-end API, and executes its applications in C++. As such it allows developers to create architectures that describe how the data gets modified, without having to implement these algorithms themselves.

Recently, TensorFlow has adopted Keras as the high-level API. The main reason to use Keras arises from its ease of learning and model building. The API was "designed for human beings, not machines", and "follows best practices for reducing cognitive load". There already exist a wide range of predefined modules – such as neural layers, cost functions, optimizers, activation functions, etc – in order to create unique models. Moreover, new modules developed by the user are simple to add amongst previously mentioned modules. Other useful advantages are its support for various production deployment options, and integration with other back-end engines (TensorFlow, CNTK, Theano, MXNet, and PlaidML), and a strong support for the use of multiple GPUs.

Keras does not perform any low-level operations, implying the computation of tensor functions, as it can rely a diverse set of back-end engines; with TensorFlow being the default. The Keras API is accessible in TensorFlow as `tf.keras`. Alternatively, it is possible to edit the active engine, for instance theano or CNTK, inside the `$HOME/.keras/keras.json` file or simply inside the Python script using the `os.environ["KERAS_BACKEND"]` property.

Furthermore, this research will be conducted in collaboration with the UMCG. While working on the neural network model, we will make sure to keep any progress saved in the cloud in case of technical failure. Google Colab is optimal for this purpose. It is a free cloud service that helps develop deep learning applications that use libraries such as Keras and TensorFlow. More importantly, Colab provides free GPU with a total of 12GB of RAM. This is essential for the training of complex neural networks.

3.2.2 Autoencoders

An autoencoder is a neural network which aims to reproduce its given input. It is comprised by a hidden layers, h, that define the result of transformations applied to the original image. Generally, the network is characterised by two parts; an encoder function $h = f(x)$ that modifies the input to a compact bottleneck, and a decoder function $r = g(h)$ that reconstructs this latent space [5]. This basic network is depicted in Figure 4. An autoencoder that simply maps $g(f(x)) = x$ everywhere is not particularly useful. Instead, they are designed to only carry out approximate reproductions of an input that relates to the training data. This way, the model targets specific aspects of the input and, as a result, learns useful features of the data.

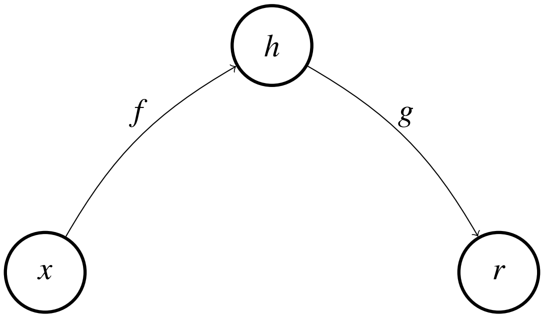


Figure 4: General structure of an autoencoder

CNNs are similar to ordinary neural networks with the difference that the weighted neurons are selected by a sliding convolution filter. Each neuron receives an input from which it learns local patterns by applying a series of non-linear transformations. This architecture maps a small convolution filter on-top of an image input, leading to a reduction of parameters in the network. Two key characteristics arise from this point. First, the patterns they discovered through the training are translation invariant, contrary to a densely connected network. Second, they can learn increasingly complex and abstract patterns through a spatial hierarchy.

Convolutional Autoencoders (CAE) are a type of CNN with the main difference in their learning procedure being unsupervised; meaning that the filters extract features with the sole purpose of reconstructing the input. Likewise, the parameters required to produce a representative activation map stays constant, regardless of the input size. For this reason they prove to be proficient with high-dimensional data. This gentle data extraction is accomplished through a sequence of steps, including the convolution layer, the reLu layer, and the pooling layer.

The convolutional layer is responsible to preserve the spatial relationship between pixels in the input image. This process is maintained by small fixed regions called feature maps. They scan through the entire input to produce a filtered output with a corresponding score, varying from some low to high values. This determines the likely match found with previously learned features. Thus, the model can extract more possible features as there are more filters. There are several methods to determine the probability of a suitable match; padding offers the possibility to drop part of the image that does not fit, or the convolution stride which controls how the filter convolves above the input.

The Rectified Linear Unit (ReLu) is a type of activation function defined as $f(x) = max(0,x)$. The introduction of ReLu can be considered as a big milestone in the deep learning community. With satisfying results, it has made its way to become the default activation function in various neural networks. This is mainly due to the mathematical simplicity of it. Generally speaking, a neural network can be optimised with ease when its behaviour is predictable, i.e. close to linear.

Finally, the role of pooling is to forcefully halve the size of a given feature map. This is realised by selecting “windows” from the input and returning its maximum value. Indeed, this process is similar to the convolutional stride, but instead of transforming fixed regions through a linear function they are hardcoded by a max tensor operation. As a result, each convolution layer will process increasingly large windows in terms of the fraction of the input they cover. Note that pooling is not the only approach to down-sample the feature maps. As briefly mentioned before, it can simply be replace by a convolutional stride without loss in accuracy on several image recognition benchmarks [22]. In particular, it appears that including pooling does not always improve performance of CNNs. With a large network, the model can learn all necessary invariances to the convolutional layer.

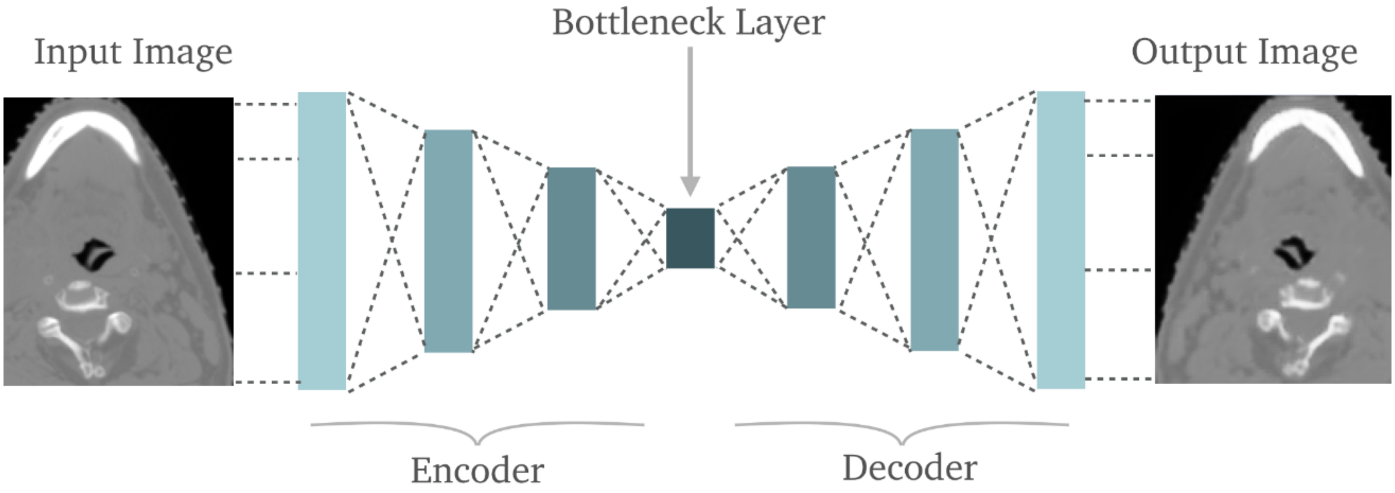


Figure 5: Basic autoencoder model architecture

With a more complete understanding of CAEs, this study devised the following architecture in order to determine cancer prognosis with CT-images. The encoder has four convolution blocks. Each block is comprised by three distinct custom convolution layers, from which the last performs a stride operation:

- A simple convolution layer with a filter size of 3 by 3.
- A batch normalisation layer, to increase the stability of the network by normalising the output of a previous activation layer.
- A ReLu activation layer.

The convolution blocks for the encoder are determined by the number of filters per layer, in this case starting at 32 up to 128 by increments of base two exponentiation. The decoder follows a similar process, with the exception of transposed convolution to reconstruct the original input. A simplification of such model is represented by Figure 5, for a more detailed architecture refer to the appendix.

Supervised machine learning maps a pair consisting of an input object and corresponding label to an output label. As such, the model is able to correctly determine the class labels for unknown instances. However, autoencoders follow an unsupervised learning that typically does not allow a suitable classification of a patients cancer prognosis. As they are constrained to reduce the input to a latent space, each layer is inclined to learn representative features to the reconstruction. This imposed limitation is also highly dependent on the application of the autoencoder; whether it is a Variational Autoencoder or a Denoising Autoencoder. Indeed, it is possibly to modify the model for feature learning and classification. Any neural network can in theory behave as a classifier by adding a `sigmoid` or `softmax` output layer. The same principle is applied to autoencoders where the decoder of a trained model is discarded and simply replaced by a `sigmoid` Dense layer, as demonstrated by Figure 6. The model is then trained for a second time with the labels all while keeping the encoder weights frozen in order to solely fine tune the output layer. This way, the features learned by the autoencoder are used to classify the input images to a specific label; in this study these refer to local recurrence, distant metastasis or death.

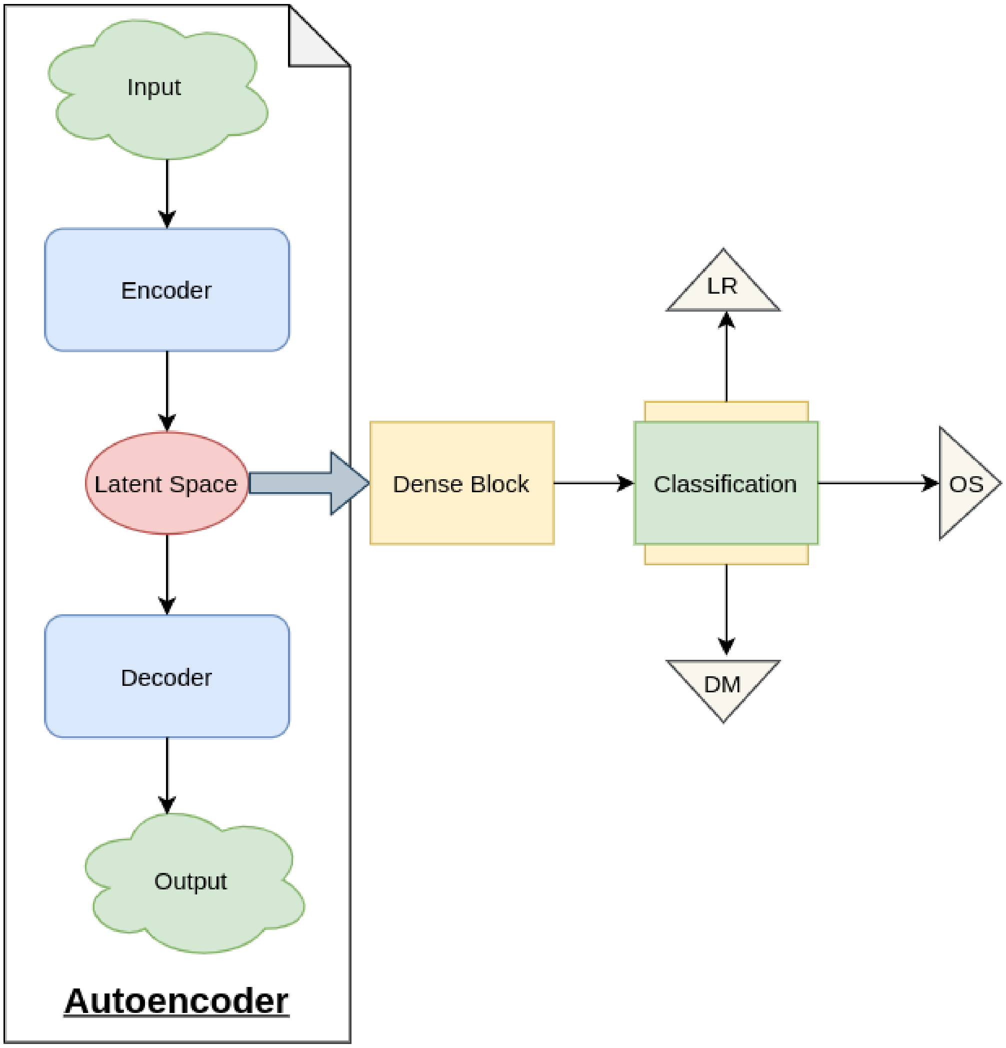


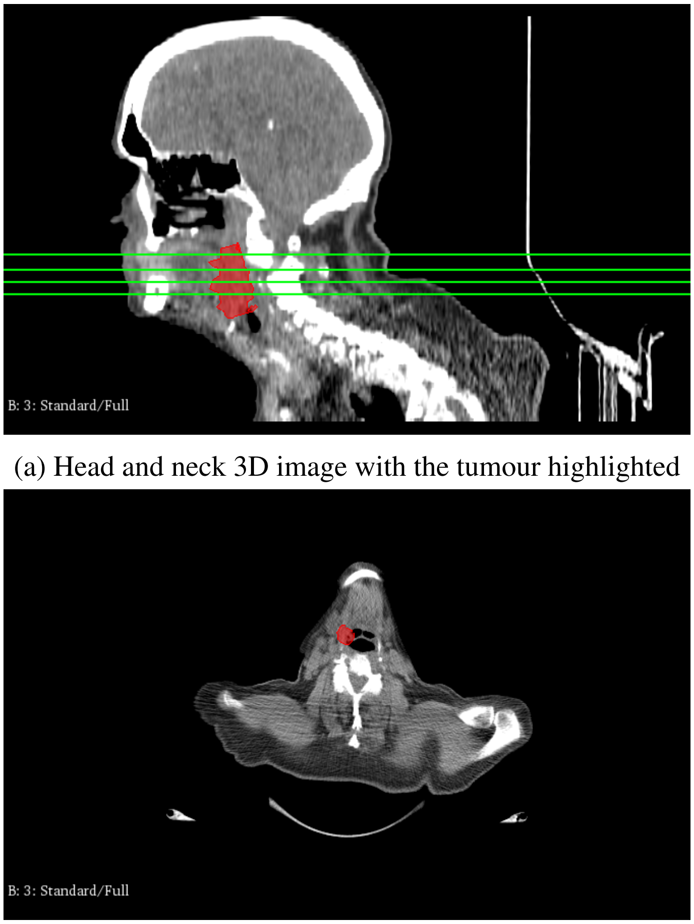
Figure 6: Modified autoencoder for classification

4 Experiments

In order to scientifically test the proposed model, some experiments that are carried out will be described. These consist of verifying the performance of the autoencoder, along with a comparison of the three different architectures. In this section the experimental set-up and evaluation methods are described.

4.1 Train and test data

Prediction errors, such as bias and variance, need to be mentioned before we discuss the decisions taken with respect to the data. Bias refers to the difference between the average that the model predicts and the proper value. Thus, a model with an important bias will typically oversimplify the model as it invests little consideration to the training data. Variance, on the other hand, is defined as the difference in performance on the training set compared to the test set. Accordingly, a high variance is considerate to the training data which in turns leads to a poor generalisation on unfamiliar data. It is therefore of importance to find a good balance between the bias and variance



(a) Head and neck 3D image with the tumour highlighted

(b) Extracted slice from the region of interest

Figure 7: Multiple region of interest located at the tumour

as to minimise the total error, i.e. a model that accurately fits the training data and can efficiently generalise the testing data.

There exist many methods that contribute to the stabilisation of a model. One approach to address the issue of a high variance due to a small training set is to ”duplicate” sections of the data. Note that this study undertakes the analysis of two dimensional input data. As a result, since the tumour covers a substantial area of the image volume, it is possible to generate ”new” patient data by dividing this ROI into equally distributed slices. This process is illustrated by Figure 7a, where each horizontal line represents a new input point such that of Figure 7b. By means of selecting multiple slices per patient, the model in question contains a more extensive foundation to train but, more importantly, the model covers a wider range of features that are representative of the tumour.

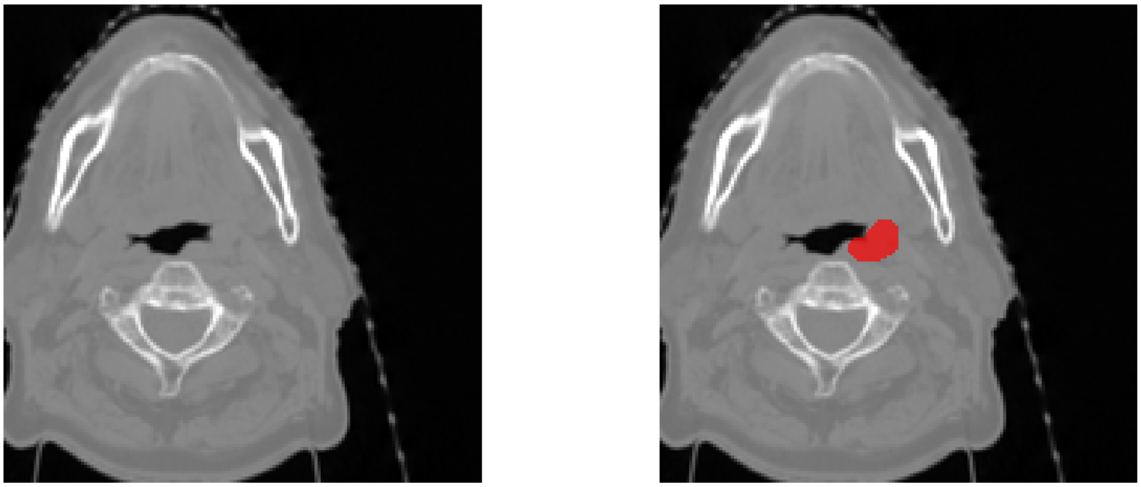


Figure 8: Pre-processed 2D image with contour mask

As briefly mentioned previously – with the current input – the model needs to learn characteristics that are not relevant for the depiction of the tumour recurrence. There are various measures that will be explored as to minimise such computation. First, more focus can be placed onto the ROI by applying a mask of the tumour contour onto the input. We anticipate the model to develop a more precise reconstruction of the tumour and subsequently learn decisive features unique to the cancer recurrence. Second, by selecting a certain window of the image, the model can then ignore some of the ”noise” surrounding the tumour. Such modifications are highlighted in Figure 8; where a window of 128 by 128 pixels is applied around the ROI.

4.2 Validation

While the model is training, the encoder learns a nonlinear mapping of the input to some latent space. The decoder, however, learns a nonlinear mapping from the latent space to the original space. The goal of an autoencoder is to minimise this loss; meaning to sample a similar distribution between the input and output. A typical loss function that minimises such reconstruction is the Mean Squared Error (MSE). The MSE evaluates the status of a predictor, a function mapping arbitrary inputs to a sample of values of some random variable, or an estimator, a function that tries to estimate some useful qualities of the population from which the data is sampled. The definition of an MSE differs depending on the nature of the problem. Such regression loss is provided by Keras and computes the mean of squares of errors between labels and predictions.

Let a sample of n data points result in a vector of n predictors. Then the MSE of the predictor can be computed as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

where Y are the observed values and $\hat{Y}$ are the predicted values. It interesting to note that the MSE predictor equation is also directly linked to a bias-variance trade-off, however this is outside the scope of this paper. In this case, the squaring is required to reduce the complexity with negative signs. The model targets to minimise this error and as a result becomes more accurate, i.e. the output produced by the model is roughly identical to the original data.

It is essential to accurately measure the reconstruction performance of an autoencoder, but in order to classify the cancer prognosis another metric is required: most classification problems use the AUC (Area Under the Curve) and ROC (Receiver Operating Characteristics). The ROC is a probability curve and AUC represents the degree of separability, i.e. the extent to which the model is able to distinguish between labels. As Figure 9 suggests, an AUC near 1 means the model can

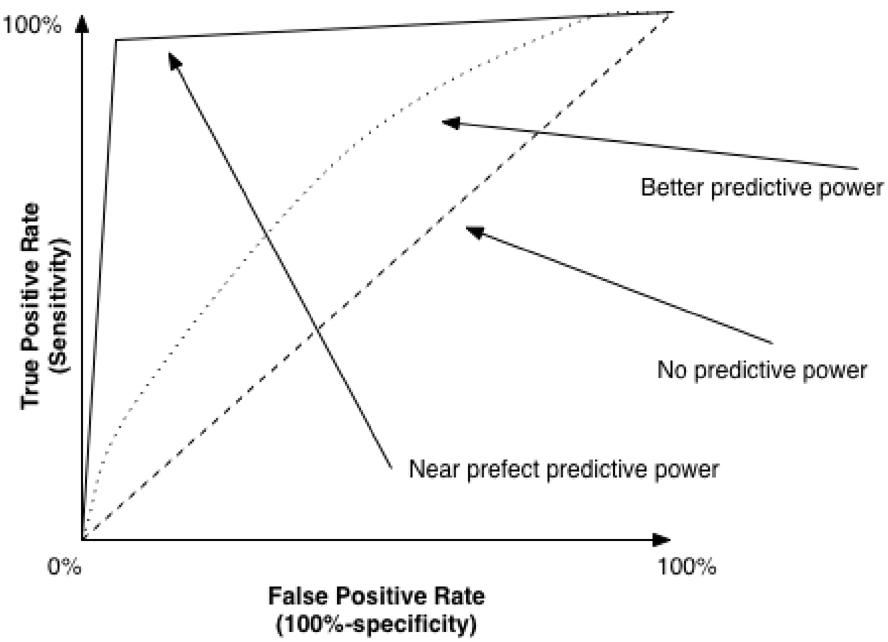


Figure 9: Significance of distinct AUC representations

properly distinguish between labels. On the other hand, with an AUC close to 0 the model is said to be reciprocating the result. At the midway point of these two extremes, the models predictive power is perfectly random.

Medical methods often use binary classification to categorise a certain disease. This can lead to two types of errors:

- **False Positive** – the neural network wrongfully indicated that the patient had the disease.
- **False Negative** – the neural network wrongfully indicated that the patient did not have the disease.

Neural networks classify in terms of the probability for the output to be positive. However, how do we determine the threshold at which the result is positive? Setting this cutoff allows the model to be more sensitive or specific. There already exist many methods that try to find the optimal cut-off point. For instance, the Youden index method implies that the threshold should be defined as the point maximising the difference between true positive rate and false positive rate over all possible cut-off points [26]. Another approach sets the optimal threshold at the point minimising the euclidean distance between the ROC and the (0,1) point [16]. Amongst many other possibilities, it is recommended that researchers should adopt a method that is clinically relevant to problem at hand.

4.3 Experimental Setup

When a machine learning algorithm is tuned for a specific problem, such as the reconstruction of input or even its classification, then the hyper-parameters of the model need adjustment to build a stable and efficient model. However, there is no decisive rule to determine these on a given problem. Typically the best values are found by trial and error.

4.3.1 Hyper-parameters

The batch size and number of epochs are two central hyper-parameters that are of interest. These are part of the stochastic gradient descent learning algorithm. It finds a set of parameters internal to the model that perform well against some loss function, as discussed this would be the MSE. Here, “gradient” refers to the computation of an error slope, and “descent” refers to this error being minimised by moving down along that slope. This algorithm is iterative. This means that the descent process develops during multiple steps, with each step revising the model parameters according to the calculated error.

The first hyper-parameter that directly influences this algorithm is the batch size. It defines the number of samples that need to be processed before the model parameters can be updated. In the most common case the batch size is more than one sample and less than the size of the training set, equal to a base two exponentiation. We have found our model to work best with a batch size of 64.

Another critical hyper-parameter is the number of epochs. This defines the number of times that the gradient descent traverses the entire training set. As such, one epoch represents a single update to the models internal parameters. Again, this value can be very volatile and is determined by intuition. With our current input and model architecture, the descent seem to find a stable minimum after 200 epochs. To be on the safe side this study will use 300 epochs to train the autoencoder.

These features are essential to build a reliable network but rely on the rather small size of our training set. In order to make the most this situation, it is possible to synthesise the data. Such data can be referred to as "made up" data, but there exist a variety of algorithms and generators to produce realistic data. This allows to expand the original data in order to aid in creating a baseline for an enhancement in the testing. We will augment the data by way of random transformations, to trick the model into seeing "new" data. This helps prevent any over-fitting and helps the model to better generalise other data. In Keras this can simply be executed via the `ImageDataGenerator` class. A few operations that will be used are mentioned below:

- **rotation_range** – a range to randomly rotate an image, 40 degrees.
- **width_shift, height_shift** – a range to randomly translate images vertically or horizontally, a fraction of 0.2.
- **shear_range** – a range to randomly apply shearing transformations, a fraction of 0.2.
- **zoom_range** – a range to randomly zooming inside pictures, a fraction of 0.2.
- **horizontal_flip** – randomly flip half of the images horizontally, set to `True`.
- **fill_mode** – strategy used for filling in newly created pixels, set to `nearest`.

At this point, the neural network is ready for training. With similar problem parameters – such as the input type and the expected output – we have defined a base to evaluate distinct models.

4.3.2 Compared networks

This study will compare the performance of head and neck cancer prognosis along different network architectures. So far we have already illustrated the model of interest that will be at the base of comparison; a deep convolutional autoencoder. The adjective "deep" simply refers to the use of multiple layers in the network which allows practical application and optimised implementation. Such architectures often generate compositional models where the complex input is modelled by fewer units. Thus the additional layers allow a deeper composition of features from the lower layers that in turn lead to a better abstraction of the data. However, by limiting the depth of layers the model is forced to learn an intelligent representation of the data. There needs to be a balance between these constraints. There exist many variants to convolutional autoencoders that have found success for different applications. With a specific dataset, it is possible to compare their efficiency and as a result determine what architecture is best suited for the problem at hand.

Autoencoders with a deep hidden layer expose themselves to the possibility of learning the identity function; meaning that the output is simply identical to the input. This data reduction is not valuable to our classification. A solution to this is the denoising autoencoder, a separate branch from the basic autoencoder. They address this drawback by randomly corrupting the input that the model then has to reconstruct. In general the percentage of input points set to null is close to 30%, but with a small dataset it is preferable to increase this rate. Figure 10 illustrate images with 40% additional noise. These are still distinguishable but the model is able to put more emphasis on the reconstruction of features.

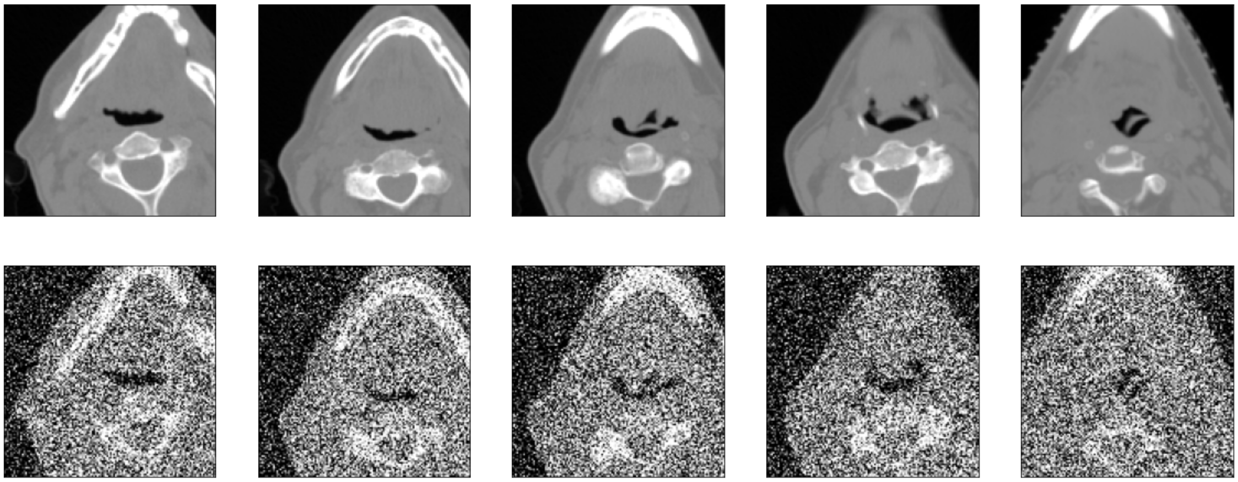


Figure 10: Autoencoder input image with 40% noise

The before mentioned dataset also provides detailed clinical attributes about every patient. To compare the achieved results with the study carried out by Vallières et al. we will expand the autoencoder classifier to connect both radiomic features through the encoder segment and the clinical data through a sequential model. Both branches are combined to a single output point, this is the resulting classification prediction. In this manner, a more in depth comparison between the extraction of radiomic and clinical features is highlighted.

5 Discussion

In this section the results of the proposed models are presented and discussed. Both the performance for image reconstruction realised by the autoencoder and the label prediction realised by the classifier are evaluated separately.

5.1 Deep Convolutional Autoencoder

As mentioned in a previous section, the performance of the convolutional autoencoder model is validated from the HMR and CHUM cohorts. With the current pre-processing strategies this results in a testing set of 409 samples.

5.1.1 Unsupervised learning

In order to assess the general performance of the previously described model we will measure the reconstruction error. This process can be brought out in distinct manners.

The MSE loss function reports how similar the reconstructed output is to the original input. Contrary to common expectations, a value of zero is not the ideal MSE. This would suggest that our model can perfectly reproduce the training data, whereas it is unlikely this would be the case for unknown data. Thus, finding a balance between a low MSE for training data – over-fitting – and a high MSE for validation data – under-fitting – is the preferred scenario. Figure 11 presents the compared loss throughout the model training.

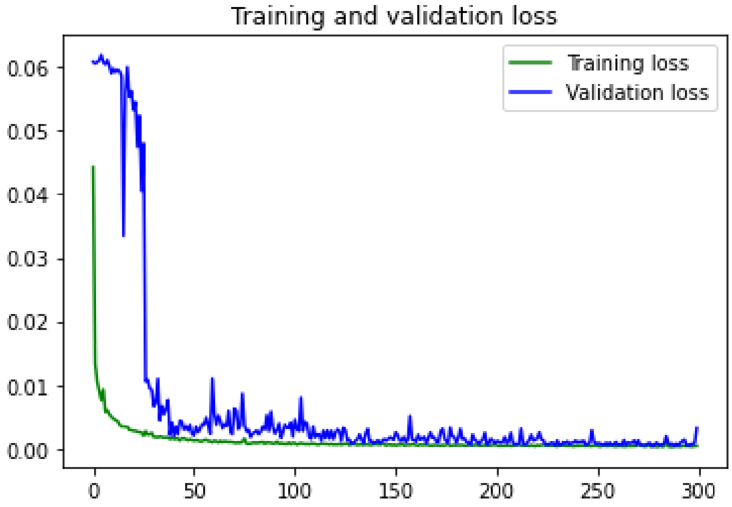
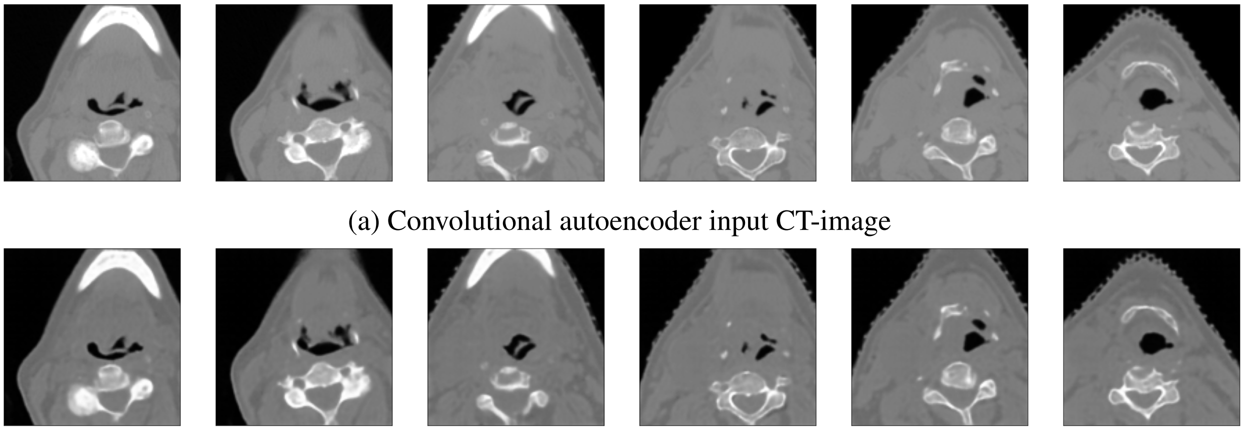


Figure 11: Convolutional autoencoder loss function throughout training

From these results it is clear that the autoencoder is able to reconstruct the input with minimal error. A more visual representation performed with new data is illustrated in Figure 12, where the input along with the respective output are aligned vertically.



(a) Convolutional autoencoder input CT-image

(b) Convolutional autoencoder output CT-image

Figure 12: Convolutional autoencoder input reconstruction

The output is very similar to the original input, with the exception of some features that appear to be less sharp. However, such evaluation is very subjective as the reconstruction of features that depict the tumour are limited by the quality of the image. A more comprehensive analysis is performed next.

5.1.2 Classification

To determine whether the model has indeed learned useful properties unique to the cancer's prognosis we undertake a meticulous classification. The graph of Figure 13 shows three ROC curves representing the performance in the prediction of local recurrence, distant metastasis, and the overall survival of the patient. The accuracy of the analysis relies on the models ability to correctly

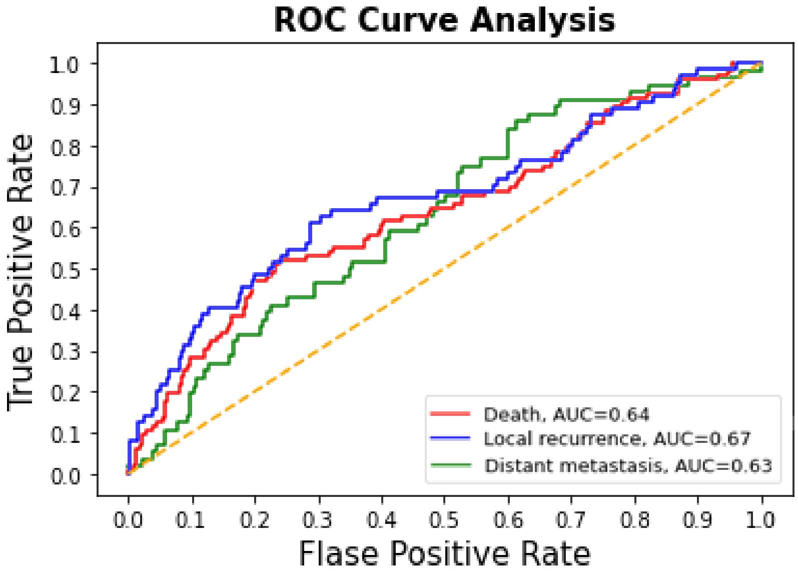


Figure 13: Convolutional autoencoder AUC measures

separate each of the labels. In this study each label was trained separately and then plotted in the same graph in order to compare their efficiency. The highest value was obtained for local recurrence prediction, with an AUC of 0.67. For distant metastasis prediction and overall survival we obtained an AUC of 0.63 and 0.64 respectively.

5.1.3 Interpretation

As demonstrated by the above results, the classification from this model is not optimal despite the promising error rate from the autoencoder. An in depth comparison from Figure 14 shows that other studies which have used the same data are encountering similar values with exception. This apparent lack in accuracy can be explained by distinct factors.

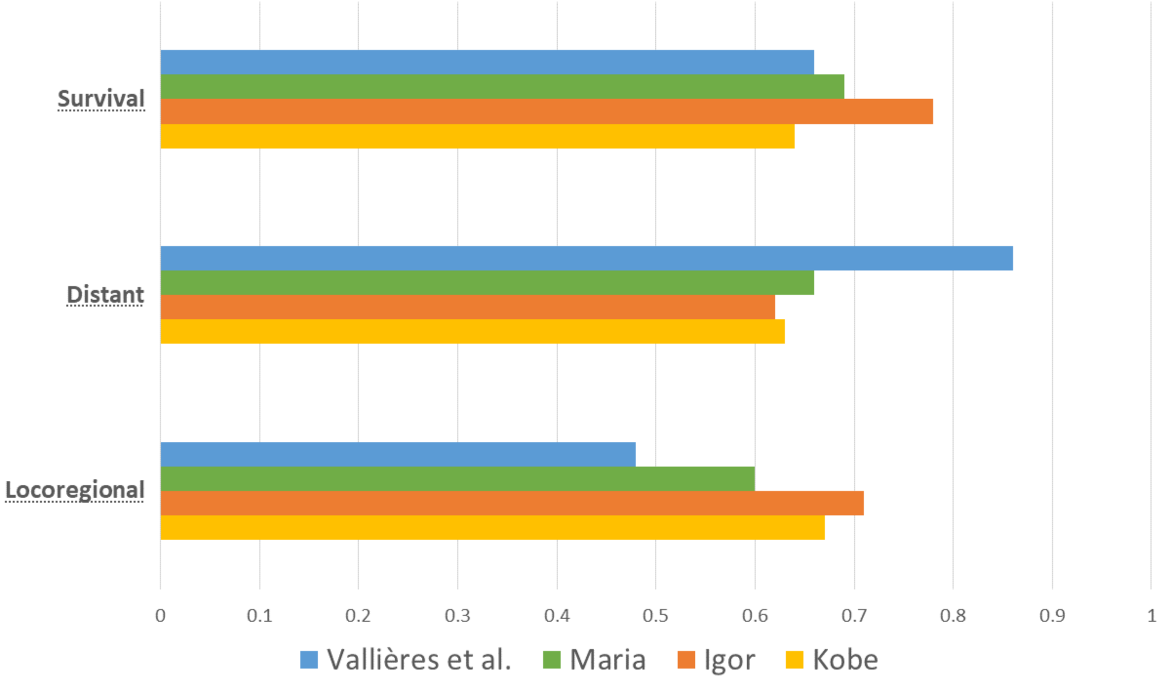


Figure 14: Extensive model comparison between studies

The first observation that becomes noticeable when studying the data is a disproportion of labels, also referred to as skewed data. Again, in our dataset only 15% of the patients developed a locoregional recurrence, 13% developed distant metastases and 19% of the patients died. Generally with a large and uniform dataset, it is expected that the deviation between the predicted and the true label will be close to zero. Such normal distribution allows a more accurate prediction rate. This is not the case with skewed data as outliers adversely affect the model's performance, notably in regression-based models. Alternatively, as seen by the results of Vallières et al, tree-based models are generally robust to such outliers. That being said, it is also possible to approximate a normal distribution with a large enough set of data. As a result, we can say with confidence that the performance of our model is deeply limited by the size and format of the data.

Furthermore, it appears that the autoencoder does not extract features unique to the cancer recurrence during the dimensionality reduction. Despite the reconstruction obtaining acceptable results, the classification suggests that the densely connected layer does not find any underlying features that allow it to correctly predict recurrence. A simple explanation to this problem would be the model does not consider specific features that represent the tumour to be relevant to the reconstruction of the input as a whole. These only represent a small fraction of the image, and as a result do not have a significant impact on the calculated error. Adding to this idea, the characteristics representing the cancer are restricted by using an autoencoder which takes 2D input. This one only has a local perspective in comparison to a 3D model, meaning it does not contain surrounding knowledge from adjacent slices. As such it becomes difficult for it to generate useful qualities of the complete tumour. This can be confirmed from the disparity in results with the two other studies that use 3D models.

5.2 Denoising Autoencoder

In an attempt to force the model to learn meaningful representations for the classification we have devised a denoising autoencoder. A similar pre-processing strategy to the previous model was followed with the exception of the input distortion, adding 40% noise to the original input. Thus, we hope this will help the model in extracting useful features which in turn will establish a more accurate classification of the recurrence.

5.2.1 Unsupervised learning

Similar to previously, the performance of this model is based on the reconstruction error calculated by the MSE. As it is demonstrated by Figure 15, the average squared difference between the estimated values and the actual value is relatively small. However, in comparison to the convolutional autoencoder, a slight over-fitting with regards to the training loss occurs. The model finds stability for the validation set after 150 epochs while the training set loss keeps gently declining.

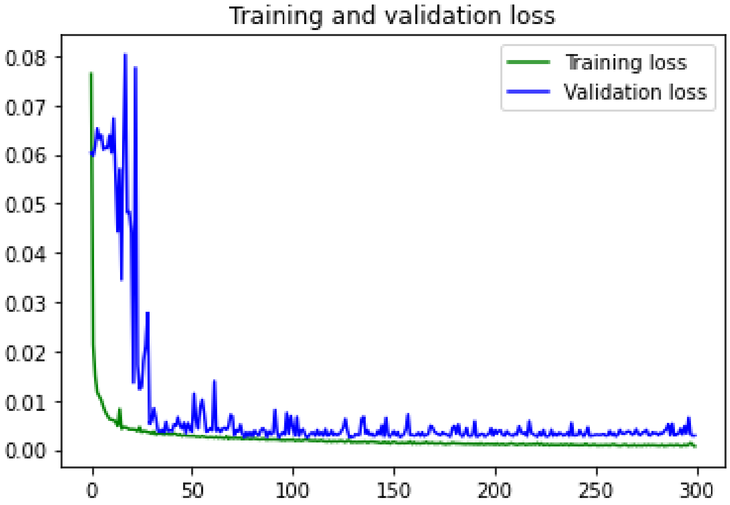
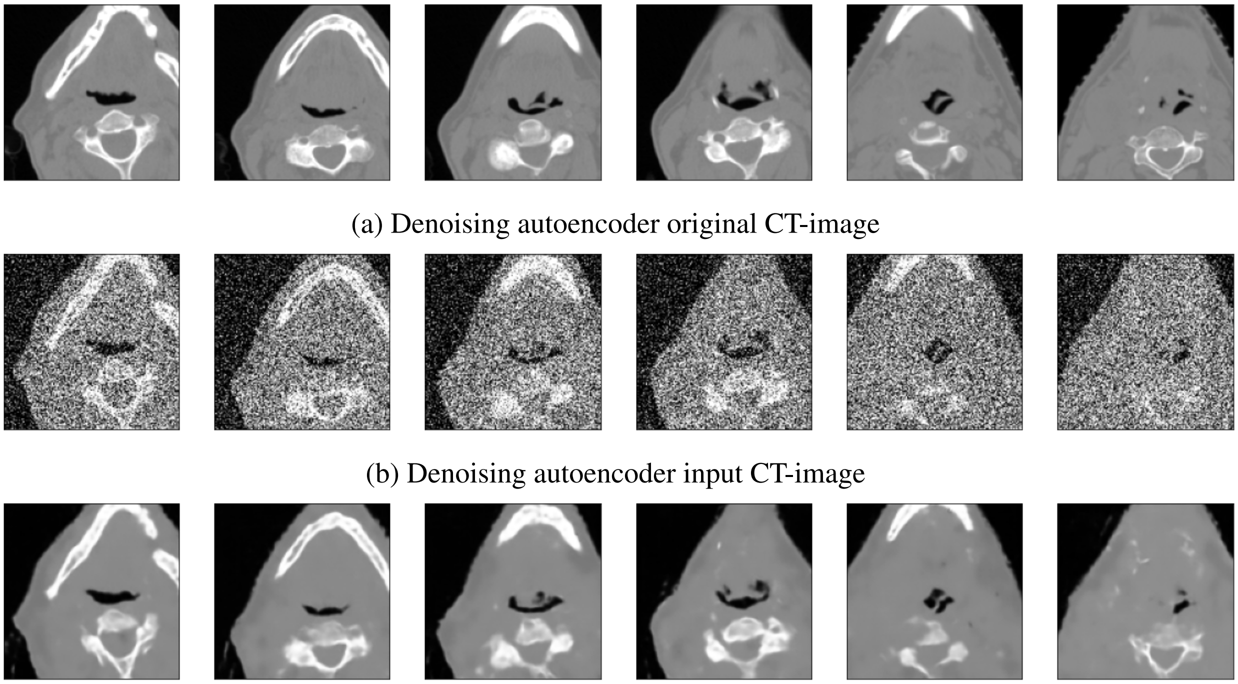


Figure 15: Denoising autoencoder loss function throughout training

In order to improve these results, a combination of different layers and number of filters were explored. Still, an architecture identical to the convolutional autoencoder provided the best image reconstruction. To better grasp the performance of the denoising autoencoder, Figure 16 displays the three processing stages: the original image, the corrupted input image, and finally the reconstructed image.



(a) Denoising autoencoder original CT-image

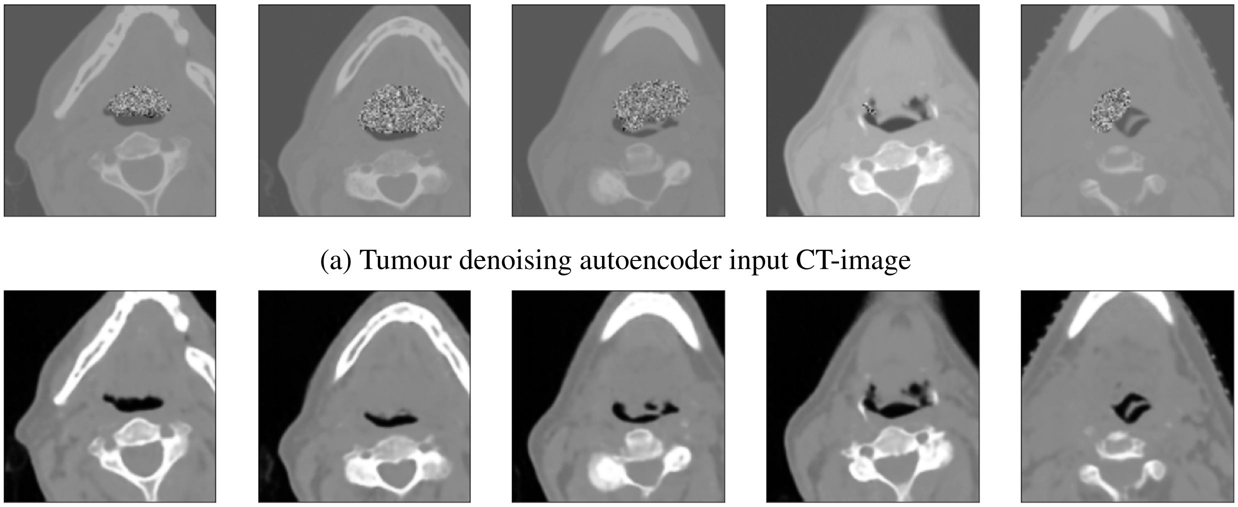
(b) Denoising autoencoder input CT-image

(c) Denoising autoencoder output CT-image

Figure 16: Denoising autoencoder input reconstruction

As such it is apparent the model is able to approximately generate an image similar to the true image. Nevertheless, by comparing the original image to the produced output it is evident that some precision is lost. Certain features are hazy or do not appear in the reconstructed image. Moreover since a large majority of the image is not relevant to the cancer's recurrence prediction, we can try to force the model in learning useful features inside a specific area, i.e. the tumour volume. As such this modification will only apply noise to the tumour, as seen by Figure 17a, with the assumption that the model will mainly focus on features unique to the cancer.

The resulting reconstruction after the training period is illustrated in Figure 17b. Indeed, there is a clear improvement with unknown data. Most of the details are apparent and in general they are sharper. But, we need to take into consideration that the reconstruction of the cancer tumour is difficult to represent though image format.



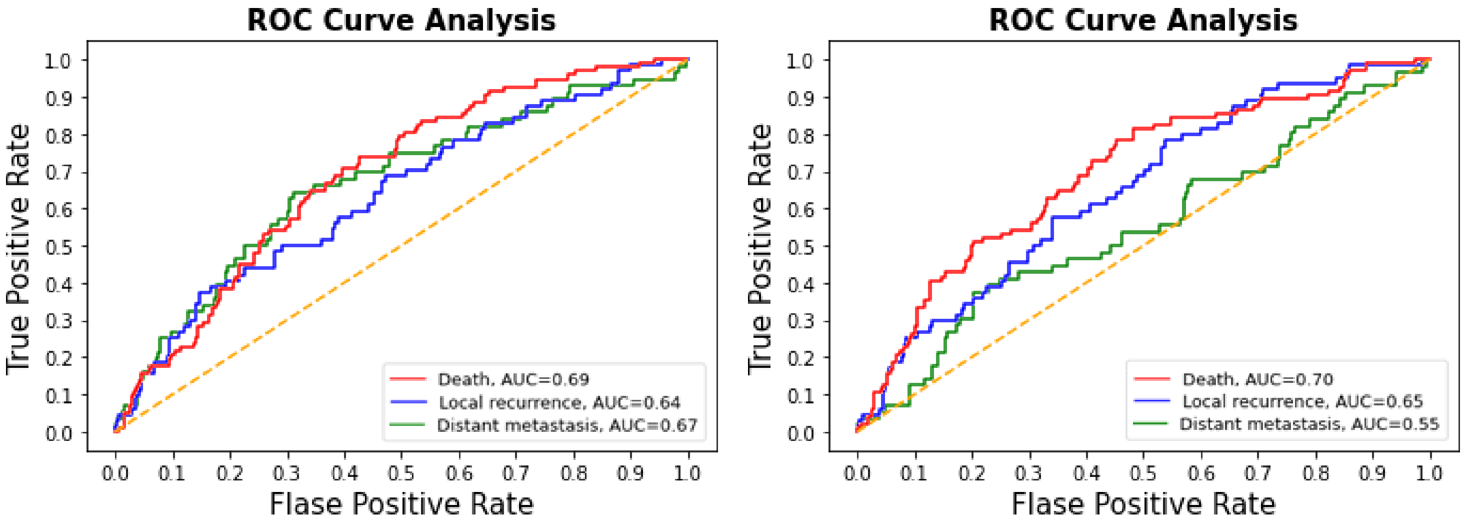
(a) Tumour denoising autoencoder input CT-image

(b) Tumour denoising autoencoder output CT-image

Figure 17: Tumour denoising autoencoder input reconstruction

5.2.2 Classification

In order to ascertain the usefulness of a learnt representation specific to the cancer's recurrence, a comparative classification of the two discussed methods will take place: first with noise added to the entire image, then only to the tumour delineation. The graph of Figure 18 shows the resulting AUC values that were generated during this classification. A slight overall improvement is noticed when applying noise to the volume. The highest value was obtained for the overall survival prediction, with an AUC of 0.69. For local recurrence and distant metastasis we obtained an AUC of 0.64 and 0.67 respectively.



(a) Tumour denoising autoencoder AUC measures

(b) Full denoising autoencoder AUC measures

Figure 18: Denoising autoencoder AUC measures

5.2.3 Interpretation

We have seen that a simple convolutional autoencoder is unable to accurately extract useful features as it tends to rely towards an obvious solution, the identity function, or uninteresting ones. By changing the reconstruction criterion, we implicitly modify the definition of a good representation to one that can partially clean corrupted input. This concept is demonstrated with the above mentioned results. In general, it appears the model is able to learn a more meaningful representation of the input for each of the applied denoising methods, regardless of a low score in Figure 18b for distant metastasis. Furthermore, by setting a focus point on the tumour it allows a more focused reconstruction with respect to the features depicting the cancer. This process is still not flawless as the area covered by the tumour is rather insignificant compared to the input image as a whole.

5.3 Multi-Input Classifier

As demonstrated by the results of Vallières et al, the combination of radiomic and clinical variables seem to greatly improve the accuracy in cancer prognosis. As an attempt to build an enhanced model, this section explores the possibility for a multi-input densely connected classifier. The convolutional autoencoder will be used as a base. Moreover, the following additional data will be used: patient age, head and neck type, T-stage, and N-stage.

5.3.1 Results

The clinical model and the radiomic model are trained on distinct data, and as a result need to be trained separately. It is only the last dense layer which combines both models. To better interpret the modified classification, the results from the clinical sequential model will first be presented, followed by the multi-input model.

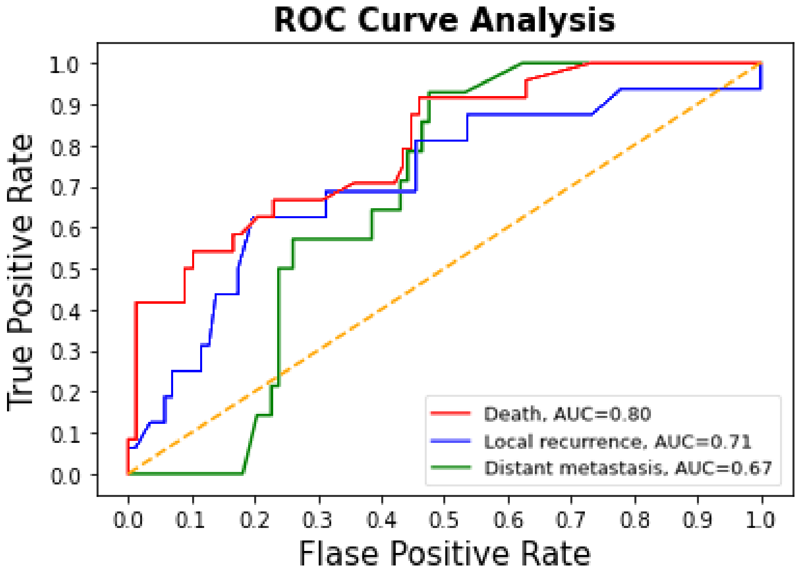


Figure 19: Clinical sequential AUC measures

As Figure 19 suggest, the results achieved with the clinical attributes on a simple sequential model are clearly proficient. They are in fact more reliable with regards to the radiomic model mentioned above. According to Vallières et al. we anticipate a substantial enhancement when combining radiomic and clinical attributes. A comparison between the results found in both studies are displayed in Table 1 below.

	Local recurrence	Distant metastasis	Overall Survival
Vallières et al.	0.62	0.85	0.73
Kobe	0.62	0.58	0.63

Table 1: Comparative table for the outcome of clinical and radiomic attributes

5.3.2 Interpretation

From the clinical attributes, a certain correlation between the chosen parameters and the cancer recurrence is manifested. However, when combining this model with radiomics, the underlying results become significantly worse. This rather unexpected outcome is thought to emerge from the contradicting features extracted from each of the two models. This discrepancy is attenuated as they are trained separately, and thus do not influence each other in learning representative features useful to both.

5.4 General Overview

The results obtained from the Convolutional Autoencoder (CAE) and both Denoising Autoencoders (DAE) are simultaneously presented in Table 2. The leading AUC measure for each risk assessment factor has been amplified. From this it is clear that the additional noise allows the model to predict the label with more confidence as it learned valuable features during the reconstruction. Further, the results seem to generally improve for each risk prediction when applying corruption to a definite delineated volume. Finally, in order to determine the ability to generalise the discussed models to various clinical institutions, the validation measures with the MAASTRO cohort is included to the table. The presented values are similar to the Quebec cohort with exception for distant metastasis. This disparity can largely be explained due to the skew existing in the data. With only 6% of patients developing a second primary tumour,rate it is simple for the model to perform well for such predictions.

	Local recurrence	Distant metastasis	Overall Survival
CAE	**0.67**	0.63	0.64
Full DAE	0.65	0.55	**0.70**
Tumour DAE	0.64	**0.67**	0.69
Clinical CAE	0.62	0.58	0.63
MAASTRO CAE	0.64	0.70	0.62

Table 2: Comparative results table for convolutional and denoising autoencoders

6 Conclusion

Cancers are known to be heterogeneous at various levels; these include genes, proteins, cells, micro-environment, tissues, and organs. However, such substances are mostly delicate to examine. Thus with the rapid development in technology, non-invasive methods allowed for quantitative imaging of these but require specific development of smart automated systems, such as deep learning networks, to extract more informative features from image-based data. Radiomics are a prominent method in the field of clinical image analysis with a promising clinical decision support system in the diagnosis and treatment of cancer. In order to apply such strategies effectively using machine learning, a certain expertise from the domain of medicine, biology, and computer science is required. As discussed to this point, deep learning has indeed the potential to improve the influence of radiomics significantly provided the available data is detailed and sufficient.

A series of cancer recurrence classification experiments were explored during this study, namely a comprehensive analysis between diverse deep autoencoder architectures. The individual performance of such models took into consideration both the reconstruction measure and the label prediction of locoregional recurrence, distant metastases, and overall survival in head and neck cancer patients. From the presented results it is possible to determine with certainty that, in the case of unsupervised learning with a 2D image input, an explicit denoising criterion helps to discover interesting features for risk assessment during the reconstruction stage. This in turn leads to an intermediate representation that is more proficient in supervised learning tasks such as classification.

7 Future Work

The opportunity to work with publicly available open source deep learning packages and data has led to a wide range application in the field of medical imaging. However, deep learning is still affected by many limitations, mainly a “black box” effect where the generated results of deep learning models lack a medical based interpretation. This restricts their use in decision making. Besides classification, it is also possible to obtain a more representative visualisation of the transformed data. Van Der Maarten et al. [13] introduced t-SNE (t-distributed Stochastic Neighbour Embedding), an algorithm to analyse high-dimensional data. Conceptually, it takes a set of points from a latent space and maps those to a lower dimension, usually the 2D plane. In general, t-SNE is flexible and can often generate a coherent interpretation from high dimensional data.

Similarly, other models such as variational autoencoders that were not included in this study allow a more accurate downstream analysis of high dimensional data. They are able to learn a representative structure of the latent space, unlike standard autoencoders that simply learn an encoding which allows them to reproduce the input with minimal error. As such, in the process of compressing the input data, they are able to reduce potential noise found in the input.

Furthermore it is common for deep learning methods such as autoencoders to suffer from overfitting; meaning the reconstruction error is insignificant on the training data. This usually leads to a exceedingly specific model that does not learn any underlying concepts found in the data. Accordingly, the model is unable to perform accurately with new data. This problem can be settled by presenting a more complete and representative training set to the model. However, such solution is often not in the grasp of researchers. A more realistic approach to improve the performance of the model would be to apply transfer learning. Such method makes use of a model pre-trained on large amounts of datasets that can assist with related problems.

References

[1] R. Bast, D. Kufe, R. Pollock, R. Weichselbaum, J. Holland, and E. Frei. *Cancer Medicine. 5th edition*. B. C. Decker, 2000.

[2] A. Dekker, S. Vinod, L. Holloway, et al. Rapid learning in practice: a lung cancer survival decision support system in routine patient care data. *Radiother Oncol*, September 2018.

[3] A. Diamant, A. Chatterjee, M. Vallières, G. Shenouda, and J. Seuntjens. Deep learning in head and neck cancer outcome prediction. *Scientific Reports*, February 2019.

[4] P. Domingos. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books, September 2015.

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

[6] L. Han and M. Kamdar. Mri to mgmt: predicting methylation status in glioblastoma patients using convolutional recurrent neural networks. *Pacific Symposium on Biocomputing*, 2018.

[7] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313:504–507, July 2006.

[8] G. Hinton and R. Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 1994.

[9] J. Kather, J. Krisam, P. Charoentong, T. Luedde, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS Medicine*, January 2019.

[10] P. Korfiatis, T. Kline, D. Lachance, I. Parney, et al. Residual deep convolutional neural network predicts mgmt methylation status. *J Digit Imaging*, August 2017.

[11] P. Lambin, E. Rios-Velazquez, R. Leijenaar, et al. Radiomics: extracting more information from medical images using advanced feature analysis. *European Journal of Cancer*, March 2012.

[12] A. Levine, C. Schlosser, J. Grewal, R. Coope, S. Jones, and S. Yip. Rise of the machines: Advances in deep learning for cancer diagnosis. *Trends Cancer*, 2019.

[13] L. Van Der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research 9*, november 2008.

[14] J. Masci, U. Meier, D. Ciresan, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. *Istituto Dalle Molle di Studi sull'Intelligenza Artificiale*, 2011.

[15] M. Palazzo, P. Beauseroy, and P. Yankilevich. A pan-cancer somatic mutation embedding using autoencoders. *BMC bioinformatics*, December 2019.

[16] N. Perkins and E. Schisterman. The inconsistency of optimal cut-points using two roc based criteria. *American Journal of Epidemiology*, 163:670–675, 2006.

[17] I. Pidik. Convolutional neural network for multi-label prediction of prognostic outcome for head and neck cancer patients. July 2020.

[18] A. Radhakrishnan, K. Damodaran, A. Soylemezoglu, C. Uhler, and G. Shivashankar. Machine learning for nuclear mechano-morphometric biomarkers in cancer diagnosis. *Scientific Report*, December 2017.

[19] M. Schabath, Y. Balagurunathan, G. Dmitry, et al. Radiomics of lung cancer. *Thoracic Oncology*, 11, February 2016.

[20] P. Simard, D. Steinkraus, and J. Platt. Best practices for convolutional neural networks applied to visual document analysis. *Microsoft Research*, 2016.

[21] American Cancer Society. How radiation therapy is used to treat cancer, December 2019.

[22] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: the all convolutional net. *International Conference on Learning Representations*, 2015.

[23] M. Stefan. Radiotherapy outcome prediction in head and neck cancer from independent 3d-cnn models. July 2020.

[24] L. Torre, F. Bray, R. Siegel, J. Ferlay, et al. Global cancer statistics. *ACS Journals*, 65:87–108, February 2011.

[25] M. Vallières, E. Kay-Rivest, L. Perrin, et al. Radiomics strategies for risk assessment of tumour failure in head-and-neck cancer. *Scientific Reports*, August 2017.

[26] W. Youden. Index for rating diagnostic tests. *American Cancer Society*, 1950.

[27] Z. Zhaoa, Y. Lib, Y. Wua, and R. Chen. Deep learning-based model for predicting progression in patients with head and neck squamous cell carcinoma. *Cancer Biomarkers*, 2020.

8 Appendix

Characteristics	HGJ	CHUS	HMR	CHUM
Male	82%	73%	76%	75%
Female	18%	27%	24%	25%
Age Range	18-84	34-88	49-85	44-90
Oropharynx	61%	72%	46%	89%
Hypopharynx	4%	1%	17%	0%
Nasopharynx	15%	6%	15%	3%
Larynx	15%	22%	22%	0%
Unknown	4%	0%	0%	8%
T1 stage	22%	9%	5%	12%
T2 stage	22%	44%	41%	43%
T3 stage	38%	30%	22%	29%
T4 stage	14%	17%	29%	8%
Tx stage	4%	0%	2%	8%
N0 stage	14%	37%	12%	6%
N1 stage	20%	11%	10%	12%
N2 stage	63%	49%	66%	69%
N3 stage	3%	3%	12%	12%
Radiation	5%	32%	17%	6%
Chemo-radiation	95%	68%	83%	94%
Locoregional	13%	17%	22%	11%
Distant	17%	10%	27%	5%
Death	15%	18%	46%	8%

Table 3: Clinical parameters from the Quebec institutions

Characteristics	MAASTRO
Male	81%
Female	19%
Age Range	44-83
Oropharynx	64%
Hypopharynx	36%
T1 stage	25%
T2 stage	23%
T3 stage	18%
T4 stage	34%
N0 stage	44%
N1 stage	12%
N2 stage	42%
N3 stage	2%
Locoregional	18%
Distant	6%
Survival	54%

Table 4: Clinical parameters from the Maastricht institution

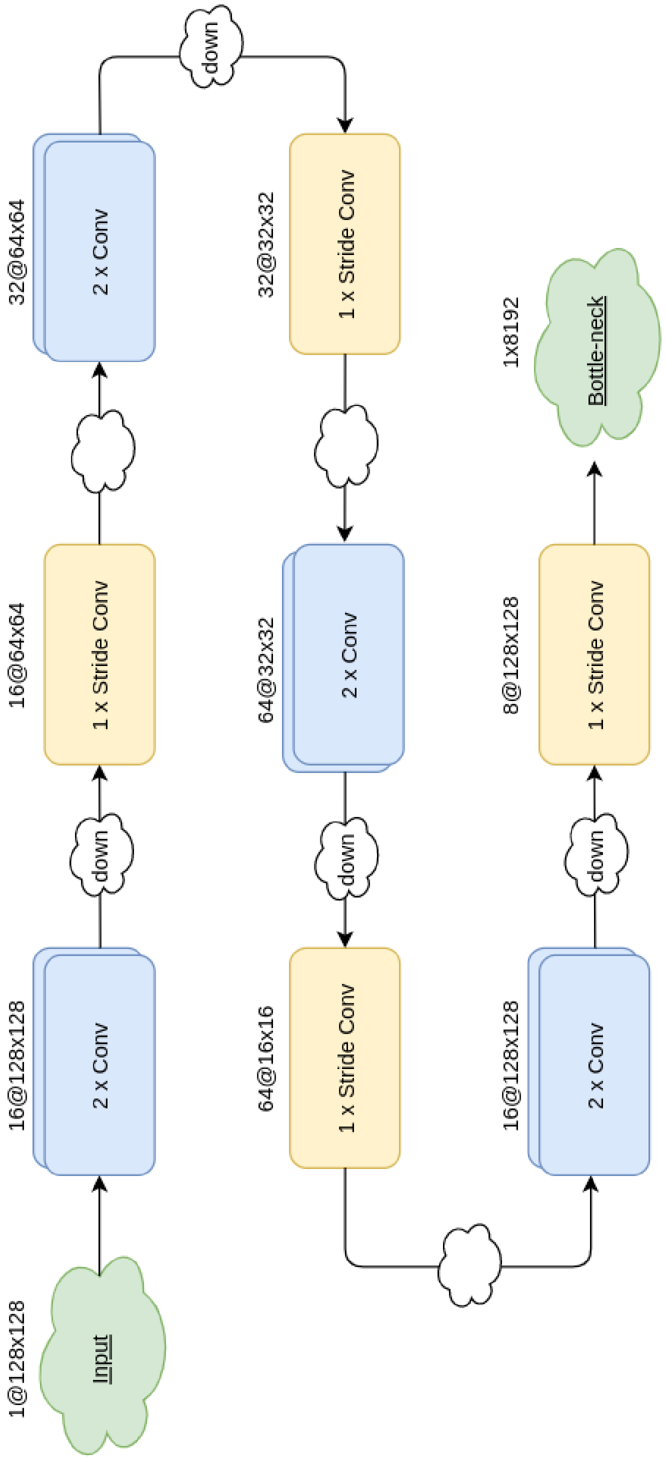


Figure 20: Encoder model architecture

This architecture describes the transformation of the input space to a reduced latent space. There are two distinct set of layers: Blue blocks – Two convolutional layers of specified filters with size [3, 3], stride of 1, and a ReLu activation function; Yellow blocks – A single convolutional layer of specified filters with size [3,3], stride of 2 for down-sampling, and ReLu activation function. These sections are alternated throughout the entire model.