Semi-invertible Convolutional Neural Network for Overall Survival Prediction in Head and Neck Cancer

Saif Eddine Khelifa, Lyes Khelladi, Miloud Bagaa, Yassine Hadjadj-Aoul

Semi-invertible Convolutional Neural Network for Overall Survival Prediction in Head and Neck Cancer

Saif Eddine Khelifa
USTHB, Algeria

Lyes Khelladi
CERIST Research Center
Algiers, Algeria

Miloud Bagaa
Aalto University
CSC, Espoo, Finland

Yassine Hadjadj-Aoul
Univ. Rennes, Inria, IRISA,
France

***Abstract*—The paper addresses the issue of overall survival prediction in head and neck cancer as an effective mean of improving clinical diagnosis and treatment planning. A new solution is proposed using semi-invertible convolutional networks. Our model exploits the 3D features of computed tomography (CT) scans to enrich the dataset used in the learning phase and thereby improve the prediction accuracy. This is achieved by designing a first architecture featuring a combination of a CNN classifier with a fully convolutional network pre-processor. The latter has been replaced in the second solution by an invertible network to deal with the memory constraints noticed in the first architecture. Obtained results showed that both architectures have led to considerable improvements in terms of prediction accuracy (0.75) compared to state-of-the-art solutions.**

I. Introduction

Head and Neck Cancers (HNC) are among the most common cancers. The HNC patients are characterized by a poor prognosis [1]. Of note, the survival rates of incurable HNC are concise. The overall survival rates of HNC cancer vary significantly according to the location and stage of the tumor. These make this type of cancer a source of utmost concern to the patients and physicians. Therefore, early diagnosis and prognosis are essential for the proper management of HNC patients [2]. Several attempts have been made for patients' prognosis estimation. For instance, the experience of the oncologists is well-positioned to estimate the prognosis of the patients [2]. However, oncologists are concerned about the possibility of making inaccurate prognoses. Additionally, this approach can be subjective and increases the risk of bias [2], [3]. The traditional statistical analysis method was outperformed by disruptive technology such as the sub-field of artificial intelligence, that is, deep machine learning in the prognosis of cancer outcomes [4], [5].

The deep learning technique leverages the improved computational power to modify the widely known artificial neural network architecture. It is a specialized machine learning technique that uses a multi-layered neural network [6]. This multi-layered architecture is able to learn the potentially complex and hidden relationships between various variables contained in the input data. Consequently, the learned relationships between these variables are used to define the output of interest, such as the overall survival of HNC patients. The deep learning technique has been reported to have significant importance in the proper management of head and neck cancer [7]. The study by Howard et al. [7] has reported the potential of deep learning techniques to identify the patients that could benefit from chemo-radiation.

The study, accomplished by Diamant et al. [8], has explored the potential of deep learning techniques to enhance the performance of traditional radiomics [8]. The latter has used computed tomography (CT) scans as the primary modality rather than other modalities, such as magnetic resonance and positron tomography. Formally, these scans are fundamental, and each modality provides different information on the tumor. CT scans are obtained from an X-ray study that produces 3D cross-sectional images of the body. Therefore, this modality can be used to reveal the tumor volume and the blood vessels that are feeding the tumor, and hence it provides more details on the tumor.

In this work, we examine the use of deep learning models applied to pre-treatment CT scans to predict overall survival in head and neck cancer. The proposed approach aims at improving the predictive accuracy for the proper management of this cancer through effective treatment and informed clinical decisions. Furthermore, we explore the potential of having 3D inputs (CT scans volume) instead of having 2D inputs (CT scans slice) to enrich the features extraction using a deep learning network by adding the spatial information between slices in the Z space. We devised two solutions for head and neck cancer survival prediction. While the first solution relies on a fully convolutional network, the second one uses the semi-invertible fully convolutional network to reduce memory utilization. The obtained results demonstrate the efficiency of each solution for achieving its design goals.

The remainder of this paper is organized as follows. First, section II briefly describes the most recent related works in the scope of predicting overall survival for HNC patients. Section III describes the employed dataset and the methods used for data filtering and pre-processing. Section IV presents the limitations of architectures based on datasets of 3D CT scans and the solutions to tackle those limitations. In section V, we present the proposed deep learning model for overall survival prediction. Section VI provides a comparative discussion considering results obtained in previous studies. Additionally, it emphasizes the clinical implications of the

overall survival prediction of head and neck cancer. Finally, section VII concludes this article.

II. Related Works

In [9], a Deep-learning convolutional neural network is used to predict radiotherapy outcomes (overall survival) on HNC cancer using 2D CT scans as inputs. In [8], the solution relies on selecting scan slices containing the most meaningful information on the tumor. This was possible by performing Gross Tumor Volume (GTV) contours that are manually delineated by experts. Using these contours, CT volumes were converted to 2D images based on the largest primary GTV lesion area slice selection. Most recently, a combination of CT and Positron Emission Tomography (PET)was most recently were employed to improve the overall survival prediction in radiotherapy treatment for HNC patients [10]. Both volumes of PET and CT scans were converted into 2D images using the largest primary GTV lesion area slice selection, which resulted in a 2D network model with two blocks. The first block is a fully convolutional network that acts as a pre-processor for image normalization using the techniques proposed [11]. The second block is the Resnext classifier inspired by the work presented in [12]. The combination of these two blocks led to notable improvements in overall survival prediction for radiotherapy treatments.

All solutions presented above rely on 2D images as input and adopt classifiers of the largest primary GTV lesion area in the slices. However, the constraints of being sealed into labeled data (GTVp GTVn) have led to a restriction in the amount of data used by the learning architecture and reduced the prediction accuracy. In fact, manual delineation of GTV contours and data labeling induce considerable costs in terms of financial budgets and time spent, which limits its application on big data size.

In our work, we explore the possibility of using the whole 3D images as data inputs in order to enrich the dataset used with additional spatial information between slices along the Z-axis. We propose a fully convolutional network-based solution that uses 3D images to predict the overall survival in head and neck cancer. The proposed approach does not assume any data labeling process via GTV contours. This approach enables us to omit the need for manual processing by experts and its related cost and potential mistakes. Nevertheless, deep learning models with 3D images may suffer from two main issues namely, depth uniformity and memory constraints. In order to mitigate these limitations, we devise a semi-invertible fully convolutional network-based solution that dramatically reduces memory utilization.

In the next section, we emphasize these two problems and discuss the possible approaches to solve them. Lessons learned from the next section will be of great help in the design of our second solution.

III. Dataset Preprocessing

A. Dataset

The dataset used in our study consists of imaging, radiotherapy, and clinical data of 627 head and neck squamous cell carcinoma (HNSCC) patients, treated at MD Anderson Cancer Center (MDACC). This dataset is publicly available through the TCGA website. Out of the 627 patients, 215 patients had computed tomography (CT) scans with non-enhanced contrast. Thus, these patients were used in our analysis. These CT scans in our dataset were provided in Digital Imaging and Communications in Medicine (DICOM) format [13]. This latter represents the standard format for medical images stored in healthcare systems.

B. Pre-processing methods

1) Filtering CT scans: As a standard, the DICOM format provides certain attributes regarding the stored scan. Accession and instance numbers are examples of these important attributes used in the filtering of the obtained CT scans. Traditionally, a single CT scanning event is indicated by one accession number. However, if the first scan is of insufficient quality, a second scan may be obtained. In this case, these two volumes from a single scanning event would have the same accession number. Controversially to the accession number that can be the same for the different scan events for the same patient, a "serial number" is unique to every scan event, even if it was done on the same patient. Furthermore, the final scans called the "original scans" can be filtered using the ImageType field in the DICOM that may equal to "Original" if the scan is an original, otherwise the high-quality scan for the same patient is selected. Therefore, to enable effective filtering of the CT scans, a volume of slices with the same serial number and ImageType ("original") was arranged to construct a proper volume. The aforementioned process was important for the proper collation of the appropriate CT scans. This process is followed by the actual pre-processing of the CT scans.

2) Pre-processing CT scans: Collected slices with the same serial number and original image type were stacked together using the *Instance Number* attribute of the DICOM. The Instance Number indicates the order of the slice in Z-space. Of note, the CT scanner's measurements are represented with Hounsfield Units (HU). The raw pixel values in the DICOM standard had undergone a linear transformation to the Hounsfield unit to enable efficient disk storage. This transformation should be reversed to obtain pixel values in Hounsfield Units once again. The DICOM standard attributes, such as *RescaleSlope* and *RescaleIntercept* are needed to linearly transform the raw pixel values stored in the DICOM pixel array into Hounsfield units. Subsequently, the pixel values were clipped to [-1000 HU, +1000 HU], as depicted in Figure 1, which represents practical lower and upper limits of the HU scale, corresponding to the radio densities of air and bone, respectively.

Further pre-processing of the scans (slices) were done by eliminating noises like patient clothes and bed through a denoising technique as depicted in figure 2.

Since the data include the volume of slices with the same serial numbers, it was important to ensure that each pixel represents a consistent volume throughout the whole data set. To achieve this, important attributes of the DICOM header

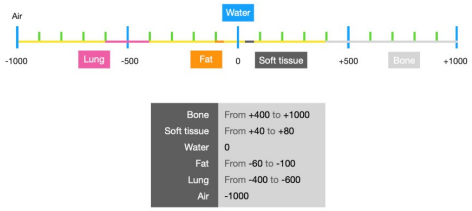


Figure 1: Hounsfield unit for each structure

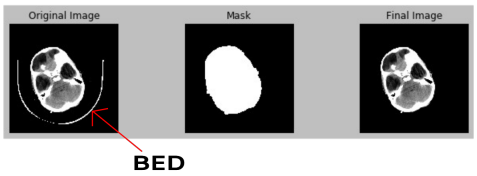


Figure 2: De-noising technique

known as *PixelSpacing* and *ImagePosition* play a key role in this regard. The *PixelSpacing* includes an x measurement and a y measurement in millimeters (X-spacing and Y-spacing). The Z-spacing (craniocaudal spacing) can be inferred by subtracting the z-position values of adjacent slices reported in *ImagePosition* patient. Taken together (x, y, z) spacing indicates the physical volume in cubic millimeters, whereby a single voxel represents a pixel in 3D space. Unfortunately, these physical distances vary according to the scan parameters, such that the physical size of each voxel is different for different patients. Furthermore, the (x, y) spacing is often different from the z-spacing for a single scan (Figure 3)

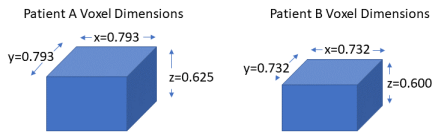


Figure 3: Voxel(3d pixel) Presentation from two different patients

To achieve consistent physical distance meaning of all pixels in the dataset, all the CT scans in our dataset were resampled to $0.8 \times 0.8 \times 0.8mm$. Finally, all volumes were normalized.

IV. Limitations and Solutions of 3D Architectures

3D architectures are heavy in terms of memory. They always suffer from memory problems during the training phase, especially with large 3D inputs like CT scans volumes. The latter also have their own problems which are depth uniformity and memory usage.

A. Depth uniformity

This is a common issue when dealing with 3D CT scans. Each patient had 3D images (CT scans) formed by a number of 2D images (slices), which is known as the depth of the 3D image. The depth can vary from one patient to another according to the CT scans. This variation imposes an additional problem in the design of the CNN architecture. That is, CNN can learn other non-important features along with the important features. Consequently, it becomes necessary to uniform the depth of the 3D images. An example of the adopted approaches for handling the depth uniformity is to randomly pick up 20 slices from the middle of each 3D image. Unfortunately, this method results in a waste in the number of slices which induces a considerable loss of invaluable information. One of the interesting approaches to handle depth uniformity while preventing the wasting of slices is the use of 3D techniques mentioned in [14]. To prepare the input data for training V-Net, the volumes of the CT scans should be normalized to the same depth. To satisfy this condition and without losing any information, CT volumes are split into smaller volumes that match the input dimensions, as depicted in figure 4.

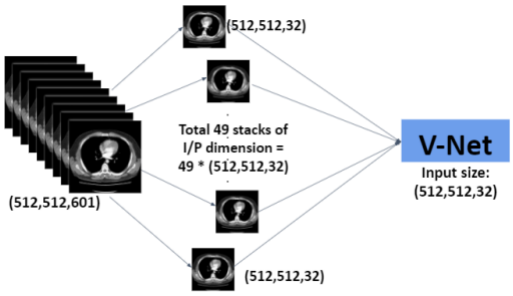


Figure 4: The 3D mini-batch technique

In this 3D technique, there are 3 variables that affect the stack or sub-volumes creation, which are: i) The CT volume that presents the number of slices in the CT scan; ii) Stack size that presents the desired number of slices in the sub-volume; iii) The overlap factor can be calculated as follows:

$$Overlap\ factor = \frac{Number\ of\ overlapping\ slices}{Stack\ size}.$$

For a CT scan having 601 slices with a stack size of 32 and number of overlapping slices as 20 (overlap factor = 0.625), we get a list of 49 stacks having dimensions $(512, 512, 32)$. The first stack will have indices from 0 to 32, which means that the first input data point for V-Net will be a CT sub volume from the 1st slice to the 32^{nd} slice. The second input data point will be a sub volume of the CT from index 12 to 44, both inclusive. As they have 20 overlapping slices, the second stack starts from the 12^{th} index and not the 33^{rd}, and so on for the rest of the stacks. For the last stack, however, the indices are $(576, 608)$, the 7 extra slices are the added paddings to keep the volume of the stack compatible with the 3D model's input dimension. During the inference using this technique, the problem of depth uniformity can be solved.

B. Memory constraint

The mini-batch technique described previously solved the depth uniformity and allowed also partially alleviate the memory problem. Indeed, By taking only small stacks, we can save the input memory consumption. However, for further improvement in terms of memory usage, invertible networks

are known to be a judicious choice. In fact, invertible networks dramatically minimize memory consumption. For this reason, we have adopted invertible UNet iUNets in our solution.

C. Basic highlight on iUNets and invertible layers

A fully invertible UNet described in [15] is similar to conventional UNets with two paths (downsampling/upsampling). The difference resides in the fact of having invertible layers instead of classical layers. Those invertible layers are then followed by a split operator to split channels by a factor of 2. The first half of the channels goes to the upsampling, while the second half is copied to the upsampling path and concatenated with upsampled layers at each level. Also, this architecture has an invertible upsampling and downsampling instead of classical upsampling and downsampling, (Figure 4 [15]).

Having invertible mapping functions will save memory usage. Instead of storing all the activation functions, only the last function will be stored. Then an inverse of this function will be calculated to obtain previous activation functions during backpropagation phase (Table I)

Table I: Comparison between U-net and Invertible U-net

Depth	Conventional	Invertible UNet	Ratio
5	3.17GB	0.85GB	26.8%
10	5.90 GB	1.09 GB	18.4%
20	11.4 GB	1.57 GB	13.8%
30	16.8 GB	2.06 GB	12.2%

1) Invertible Layer(Additive coupling): Additive coupling is simple invertible layers that represent the main components of iUNets. They are conceptually similar to the residual layer and their inverse mapping can be computed just as fast as their forward mapping. These layers are employed to compute features within each specific resolution of the iUNet, as depicted in figure 5.

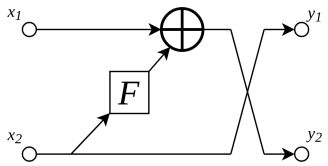


Figure 5: Additive coupling layers as implemented in this library. The input and output are split into two groups of channels. Here, the computational block F can be any sequence of e.g. convolutional layers with normalization. Note that the roles of x1 and x2 are reversed afterward.

The main idea behind additive coupling is splitting the activation's x (input) and y (output) into two groups of channels. i.e (x_1, x_2) for x and (y_1, y_2) for y. This is in order to define the invertible mapping xy via $y_1 y_2 = x_2 = x_1 + F(x_2)$, where the F can be a simple arbitrary mapping that maps from the space of x_2 to the space of x_1. In practice, F can be as simple as a sequence of convolutional layers with batch normalization. This function will be defined in next section.

V. Methodologies

In this section, two methods were used to predict overall survival using 3D input data (CT scan volumes) while eliminating the constraint of having limited labeled data (GTV contours).

A. Fully Convolutional Network

As mentioned earlier, the employed 3D architecture is inspired by the work of Le et al. [10] using V-Nets (fully convolutional network) [16] as a pre-processor. Drozdzal et al. [11] have previously shown that the V-net technique acts as an image normalization on medical images for liver tumor segmentation, as depicted in Figure 6. Then, it is followed by a fully convolution classifier inspired by the ResNeXt architecture [12], as depicted in Figure 7. This architecture was implemented in 3D format to make use of 3D inputs and to have more information in 3D space as illustrated in Figure 8.

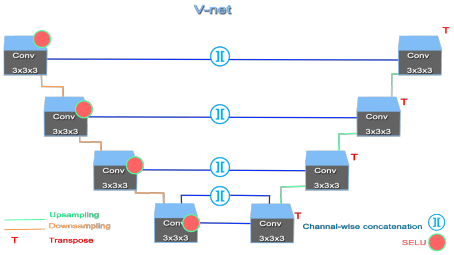


Figure 6: Preprocessor 3D unet (V-net)

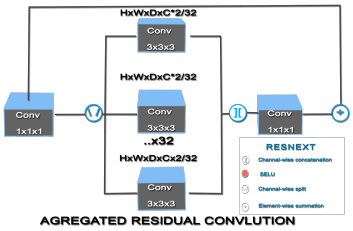


Figure 7: 3D Resnext

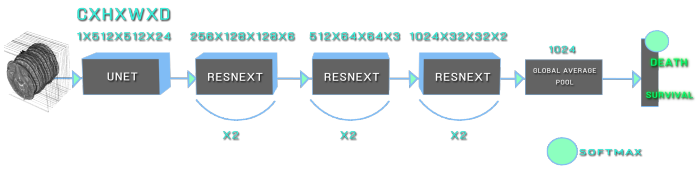


Figure 8: Fully convlutional architecture

The inputs of this architecture were a 3D CT volumes with 3D technique as mentioned in [14] to deal with depth uniformity problem. However, this architecture suffers from the memory problem since 3D UNet (V-NET) [16] is a heavy network. In order to solve this problem, a part of the 3D UNet was replaced by an Invertible UNet to reduce the memory consumption, which led us to a Semi-Invertible Fully convolutional network.

B. Semi-Invertible Fully Convolutional Network

In this study, a new architecture called the Semi-Invertible Fully Convolutional Network is suggested. The latter is more efficient in terms of memory utilization. It is built through replacing the pre-processor block with an inverted pre-processor (3D Invertible UNet) with 4 invertible downsampling blocks

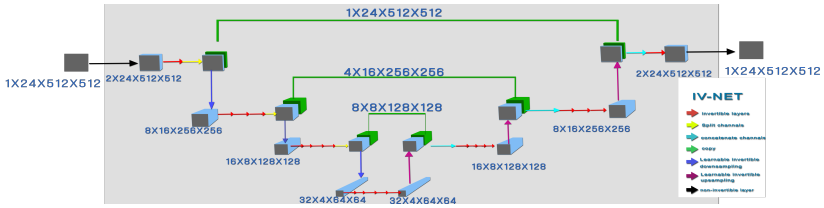


Figure 9: The invertible pre-processor (3D invertible UNet) with custom additive coupling (invertible layer)

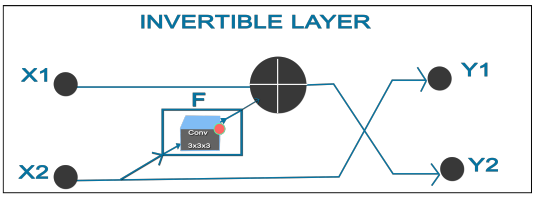


Figure 10: Custom invertible Layer

followed by 4 invertible upsampling blocks, as shown in Figure 9. Each block consists of a number of invertible layers at each level. Each invertible layer includes a coupling function F, which is a sequence layer that consists of 3x3x3 convolutional layer followed by SELU function for regularization purposes (Figure 10). Then, it is followed by a classifier that consists of 3D Resnext (Figure 7) blocks with two 3x3x3 bottleneck layers, with a filter growth factor of 2, and cardinality of 32 and residual connections. Each connection is finalized by global average pooling [12], and fully connected layers are used to output the binary survival class, as depicted in figure 11.

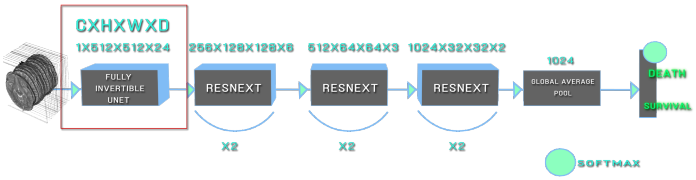


Figure 11: The Semi-Invertible architecture

VI. Results and Discussion

The prediction of the overall survival of HNSCC patients was considered as a classification problem in this study. For the classification task (0:survival , 1:death), different parameters were taken for each proposed method as shown in Table II.

Table II: Models parameters

Network	Mini-batch	Batch	Learning rate
Model (Invertible UNet)	24	1	0.0002
Model (Classic UNet)	1	1	0.0006

The proposed architecture is implemented using Python and Pytorch library [17].

A. Data Augmentation and Preparation

Remarkably, our target output variable, i.e. 0:survival , 1:death was imbalanced. In our case, the number alive was significantly higher than the number of death cases. While the number alive is 458, the number of death cases is 169. To handle this incidence of imbalanced dataset, oversampling technique is used.

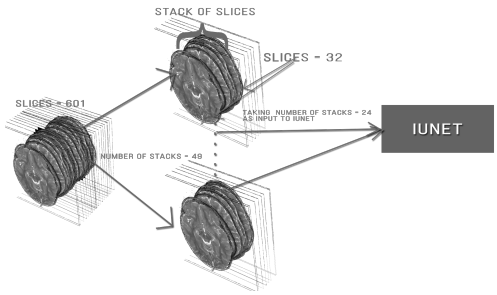


Figure 12: Choosing the right mini-batch

In an attempt to handle the data imbalance through an oversampling technique, a rotation of 20 degrees and random flip with a probability of 50% were applied. This method allowed to fill the gap between the 2 targets, such that each class has 458 cases.

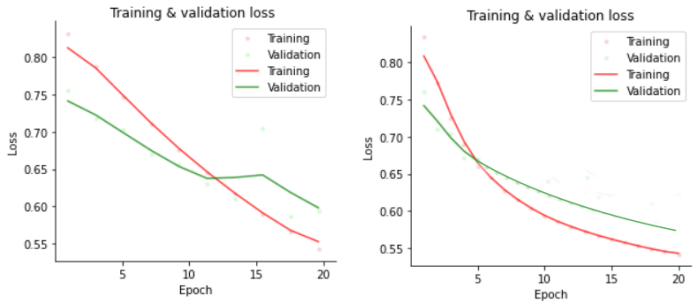
After augmenting the input dataset, an important phase was to apply the mini-batch 3D technique discussed earlier, as illustrated in Figure 4. This last was performed in order to converge towards the optimal value of the mini-batch that has undergone several changes during the training figure 12.

At this stage, each patient has a CT scan volume ranging between 300 and 750 2D slices (after re-sampling). By applying the 3D technique formula on the minimum volume, 24 sub-volumes were extracted with depth equals to 32, and this number of sub-volumes was taken as *"mini-batch = 24"*. Consequently, the model can accept 24 sub-volumes of depth 32 and resolution 512×512 at once (feed-forward pass). Based on this value (i.e., Mini-batch = 24), having a patient's size volume higher then 300 can give a higher number of sub-volumes (i.e., higher then 24 sub-volume).

For example, the volume size of 700 slices, with 512×512 resolution produces 57 sub-volumes with a depth of 32 and a resolution of 512×512. In order to keep the mini-batch at 24 sub-volumes, the 48 is divided by 2, which results 2×24 sub-volumes. Then, the network has a 24 sub-volumes two times (two forward passes). On the other hand, having a volume size that is not divided by 24 can be processed by taking the quotient of the division and multiplying it by 24. i.e 24 x quotient this will result in how many sub-volumes should be taken from the middle of the sub-volumes.

B. Training Architectures and effect of invertible pre-processor

In both Fully Convolutional Architecture and Semi-invertible Architecture, the performance of the network is



(a) Loss curves for fully convolutional network

(b) Loss curves for Semi-invertible network

Figure 13: A loss comparison between the fully convolutional and semi-invertible solutions

evaluated using the accuracy metric defined as follow:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ no\ of\ predictions}$$

with cross-entropy loss function:

$$Cross\text{-}entropy = -\sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j}\log(p_{i,j})$$

During the training phase, both architectures are used in order to demonstrate the effect of adding invertible UNet on the overall survival accuracy, as shown in Table III.

Table III: Observing the effect of invertible UNet on the accuracy of our model to predict overall survival

Network	Memory usage(GB)	Mini-batch	Accuracy
Model (Invertible UNet)	9.902	24	0.75
Model(Classic UNet)	14,756	1	0.70

Despite of the accuracy of the Fully Convolutional architecture that equals to 0.70 (Figure 13(a)), the model suffered from performance and memory constraints. Using the Semi-invertible UNet , as depicted in Table III, led to an increase of the mini-batch size to 24. It also led to a significant improvement in the accuracy to 0.75 (i.e., 75%), as depicted in Figure 13(b).

Comparing the loss curve of both networks, we noticed that the Semi-invertible network loss was slightly enhanced, compared to the first architecture. However, the graph is not quite perfect because of the unrepresentative data points in the data validation set. In addition, the data augmentation phase (oversampling) was not sufficient to cover the huge gap between the labels (alive, dead).

VII. Conclusion

The present study examined the use of a deep learning model for the estimation of overall survival in head and neck cancer patients. In this study, we explored the application of semi-invertible UNets for survival prediction in head and neck cancer. The overall accuracy obtained by our solution demonstrated the potential of some emergent deep learning techniques in building models to assist clinicians in the proper treatment of HNC patients. Nevertheless, the use of deep learning architectures for overall survival estimation in head and neck cancer are still facing multiple challenges that are worth considering. For example, it is important to empower the developed models with additional techniques that help in explaining and interpreting the predicted results, which will foster their adoption in actual clinical settings.

References

[1] Hoesseini A, Offerman MPJ, van de Wall-Neecke BJ, Sewnaik A, Wieringa MH, Baatenburg de Jong RJ. Physicians, "clinical prediction of survival in head and neck cancer patients in the palliative phase." *BMC Palliative Care*, 2020.

[2] Alabi RO, Mäkitie AA, Pirinen M, Elmusrati M, Leivo I, Almangush A., "Comparison of nomogram with machine learning techniques for prediction of overall survival in patients with tongue cancer," *Int J Med Inform*, vol. 145, no. 104313, 2021.

[3] Vlaev I, Chater N., "Game relativity: how context influences strategic decision making. j exp psychol learn mem cogn." vol. 32, no. 131, 2006.

[4] Tseng W-T, Chiang W-F, Liu S-Y, Roan J, Lin C-N, "The application of data mining techniques to oral cancer prognosis." *Medical Systems*, vol. 39, 2015.

[5] Zhu L, Luo W, Su M, Wei H, Wei J, Zhang X, et al, "Comparison between artificial neural network and cox regression model in predicting the survival rate of gastric cancer patients," *Biomedical Reports*, pp. 757–760, 2013.

[6] Chu CS, Lee NP, Ho JWK, Choi S-W, Thomson PJ., "Deep learning for clinical image analyses in oral squamous cell carcinoma:a review," *JAMA Otolaryngol Head Neck Surg*, 2021.

[7] Howard FM, Kochanny S, Koshy M, Spiotto M, Pearson AT., "Machine learning–guided adjuvant treatment of head and neck cancer," *JAMA Network Open*, 2020.

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

[9] Martin Vallires, Emily Kay-Rivest, Lo Perrin, Xavier Liem, Christophe Furstoss, Nader Khaouam, Phuc Nguyen-Tan, Chang-Shu Wang, and Khalil Sultanem., "Data from head-neck-pet-ct," 2017. [Online]. Available: https://wiki.cancerimagingarchive.net/x/24pyAQ

[10] William Le, Francisco Perdigón Romero, Samuel Kadoury., "A normalized fully convolutional approach to head and neck cancer outcome prediction," p. 6, 2020.

[11] Michal Drozdzal, Gabriel Chartrand, Eugene Vorontsov, Mahsa Shakeri, Lisa Di Jorio, An Tang, Adriana Romero, Yoshua Bengio, Chris Pal, and Samuel Kadoury., "Learning normalized inputs for iterative estimation in medical image segmentation." vol. 44, pp. 1–13, 2018.

[12] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He, "Aggregated residual transformations for deep neural networks," 2016.

[13] [Online]. Available: https://www.dicomstandard.org/

[14] Abhishek Shivdeo, Rohit Lokwani, Viraj Kulkarni, Amit Kharat, Aniruddha Pant, "Comparative evaluation of 3d and 2d deep learning techniques for semantic segmentation in ct scans," p. 9, 2021.

[15] C. Etmann, R. Ke, and C. Schönlieb, "iunets: Fully invertible u-nets with learnable up- and downsampling," *CoRR*, vol. abs/2005.05220, 2020. [Online]. Available: https://arxiv.org/abs/2005.05220

[16] Fausto Milletari, Nassir Navab, Seyed-Ahmad Ahmadi , "V-net: Fully convolutional neural networks for volumetric medical image segmentation," 2016.

[17] [Online]. Available: https://pytorch.org/