

Towards Tumour Graph Learning for Survival Prediction in Head & Neck Cancer Patients

Ángel Víctor Juanco-Müller^{1,2}[0000-0003-2724-7903], João F. C. Mota²[0000-0001-7263-8255], Keith Goatman¹[0000-0003-1379-1860], and Corné Hoogendoorn¹[0000-0002-4914-9936]

¹ Canon Medical Research Europe Ltd., Edinburgh, UK

victor.juancomuller@mre.medical.canon

² Heriot-Watt University, Edinburgh, UK

Abstract. With nearly one million new cases diagnosed worldwide in 2020, head & neck cancer is a deadly and common malignity. There are challenges to decision making and treatment of such cancer, due to lesions in multiple locations and outcome variability between patients. Therefore, automated segmentation and prognosis estimation approaches can help ensure each patient gets the most effective treatment. This paper presents a framework to perform these functions on arbitrary field of view (FoV) PET and CT registered scans, thus approaching tasks 1 and 2 of the HECKTOR 2022 challenge as team `VoKcOw`. The method consists of three stages: localization, segmentation and survival prediction. First, the scans with arbitrary FoV are cropped to the head and neck region and a u-shaped convolutional neural network (CNN) is trained to segment the region of interest. Then, using the obtained regions, another CNN is combined with a support vector machine classifier to obtain the semantic segmentation of the tumours, which results in an aggregated Dice score of 0.57 in task 1. Finally, survival prediction is approached with an ensemble of Weibull accelerated failure times model and deep learning methods. In addition to patient health record data, we explore whether processing graphs of image patches centred at the tumours via graph convolutions can improve the prognostic predictions. A concordance index of 0.64 was achieved in the test set, ranking 6th in the challenge leaderboard for this task.

1 Introduction

Tumours occurring in the oropharyngeal region are commonly referred to as head and neck (H&N) Cancer. In 2020 they were the third most commonly diagnosed cancers worldwide [1]. To inform the difficult decisions that oncologists often have to make, prognosis estimation has been shown to result in better treatment planning and improved patient quality of life [2]. Therefore, automatic lesion segmentation and risk score prediction algorithms have the potential to speed up clinicians workloads, enabling them to treat more patients.

The HECKTOR challenge was conceived [3, 4] to advance the task of automatic primary tumour (GTVp) segmentation and prognosis prediction. Since

the first edition in 2020, the dataset has increased from 254 to 325 cases in 2021 [5], and up to a total of 883 cases in the 2022 edition [6]. Other characteristics of the present release are the lack of region of interest (RoIs) and the inclusion of secondary lymph nodes (GTVn) as segmentation targets.

This paper describes a framework for tumour segmentation and prognosis prediction consisting of three stages. First, a localization model finds the neck region in the input scans (§2.1). Then, to obtain segmentation masks for task 1, we train a u-shaped convolutional neural network (UNet) [7, 8] to distinguish between tumour and background, and a support vector machine (SVM) to predict the tumour type and discard false positives (§2.2), resulting in the semantic segmentation output for task 1, which achieves an average Dice score (DSC) of 0.57 in the test set. Finally, we explore combinations of the deep multi task logistic regression (MTLR) model, featuring CNNs and graph convolutions networks, and the Weibull accelerated failure times (Weibull AFT) method to predict the prognosis metric, which is the *relapse free survival* (§2.3), resulting in a concordance index of 0.64 in the test set.

The experimental implementation is detailed in §3, results are presented in §4, and a discussion of our findings is provided in §5.

2 Materials & Methods

This section presents the three main stages of our framework: localization §2.1, segmentation §2.2 and survival prediction §2.3, depicted in Fig. 1.

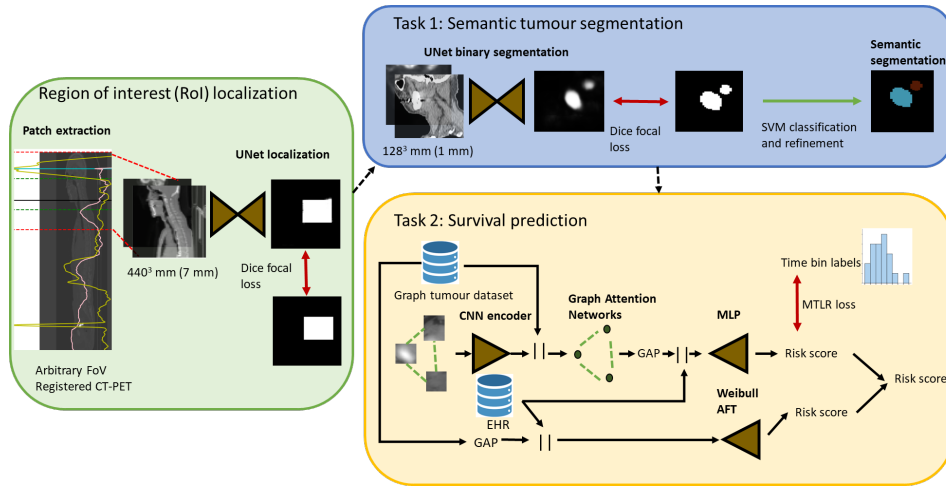


Fig. 1. The three main stages of the proposed framework. GAP stands for global average pooling, MLP for multi layer perceptron, and || refers to concatenation.

2.1 Localization

First, we extract 440^3 mm patches of the head and neck region from the arbitrary FoV PET-CT scan by analysing the CT and PET mean slice intensity along the z-axis. The brain is detected by a peak in the PET signal and the neck by an abrupt drop of the CT value. To avoid false positives caused by peaks of the PET signal in other regions of the body (e.g., bladder) we restrict the landmark search only to the first 250 mm starting from the head, as depicted in Fig. 2.

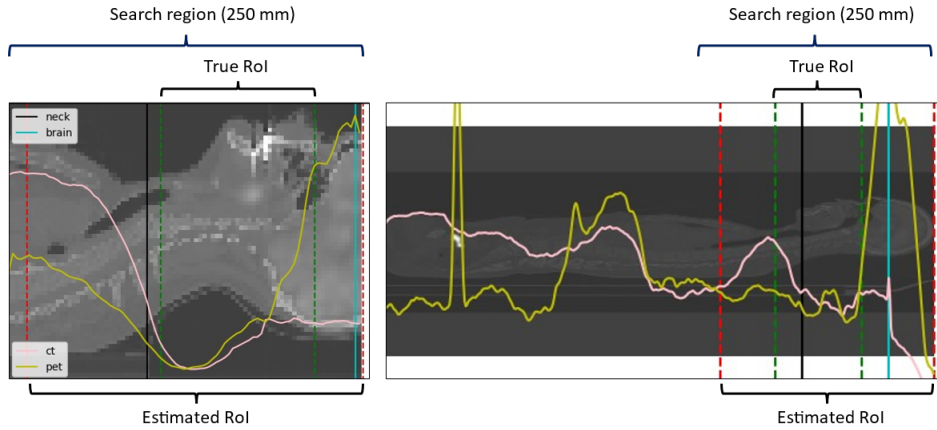


Fig. 2. Patch extraction for two different FoV cases. The dotted red lines are the inferred bounds, whereas the green ones correspond to the ground truth location of the RoI.

The resulting patches are then resized to a 64^3 mm size with trilinear interpolation for the images and nearest neighbours for the reference bounding box masks, which are obtained from the ground truth tumour segmentations. A 3D UNet [7, 8] is then used to segment the latter by minimizing the sum of Dice [20] and focal [22] losses (Fig. 3), achieving a Dice score in the validation set of 0.72. This results in a model with 3 layers, each comprising convolutional blocks, ReLU activations, and instance normalization [21]. From one layer to the next, we double the number of channels and reduce the spatial dimensions by half with max pooling.

2.2 Segmentation

For the segmentation task, we first apply a 3D UNet [7, 8] based on the model presented by the top ranked teams in 2020 [9] and 2021 [10]. The UNet has 5 levels of depth, without exceeding 320 channels, and uses residual squeeze and excitation blocks [8]. The loss function was the same one used for localization.

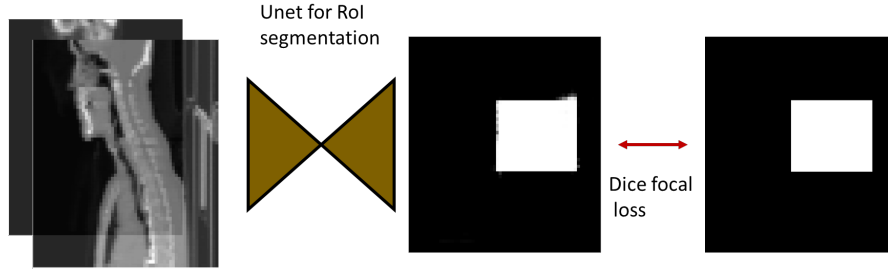


Fig. 3. ROI segmentation with UNet. Based on the bounds inferred in the previous step, the images are cropped to a common FoV and a UNet is trained on a low-resolution dataset to segment the target ROI bounding box.

Because a multi-class segmentation model performed poorly, we opted instead to use the UNet for binary segmentation (tumour-background), and a traditional classifier to infer the tumour type. We tried different algorithms and a support vector machine (SVM) with radial basis functions was chosen as it yielded the best performance. The input features were: *tumour centroid* and *bounding box coordinates*, *Euler number*, *extent*, *solidity*, *filled area*, *area of the convex hull*, *area of the bounding box*, *maximum Feret diameter*, *equivalent diameter area*, *eigenvalues of moment of inertia* and *minimum, mean and max values of CT and PET intensities*.

The SVM classifies the tumours into three possible classes: background (to discard false positive predictions), primary tumour (GTV_p) and lymph node tumours (GTV_n). The input features were extracted with the Scikit-Image library [12], and the SVM implementation is provided by the Scikit-Learn [13] library with default parameters. The overall pipeline is depicted in Fig. 4.

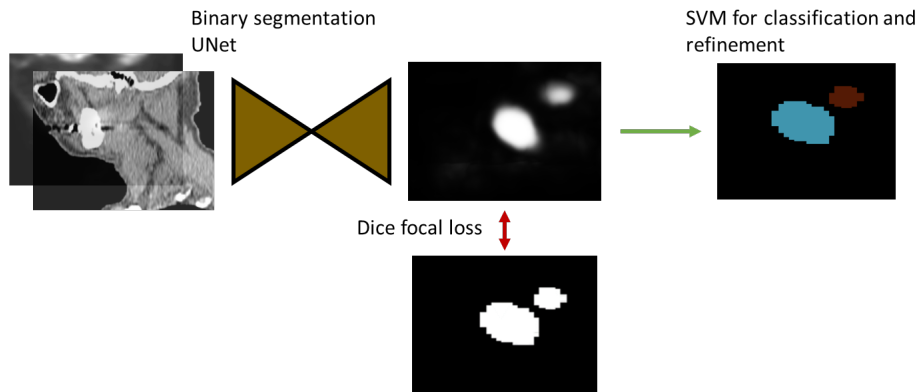


Fig. 4. Semantic tumour segmentation. First a UNet segments the tumours from the background and then an SVM classifier predicts the tumour type.

2.3 Survival Prediction

We first use simple models to select the most relevant tumour features for survival prediction, and then explore deep learning models that process such features together with the images and clinical data.

Calibration Experiments. We considered the *Cox proportional hazards* (Cox PH) [15] and *Weibull accelerated failure times* (Weibull AFT) [14] methods. We fitted these models on only electronic health record (EHR) data and both EHR and different combinations of the features used for tumour classification with SVM. For cases with several tumours, we considered the mean value of the features. We also included the number of tumours as an additional feature. After experimentation we identified *tumour centroid*, *mean CT and PET*, *max CT intensities*, and *number of tumours* as the combination yielding the best improvement of survival prediction in the validation set.

Among the EHR variables, there was missing information for some patients regarding alcohol and tobacco usage, performance status in the Zubrod scale, presence of human papillomavirus (HPV) and whether the patient has undergone surgery or not. Because all these variables are non-negative, we assigned the value -1 for the missing cases. This resulted in better performance than simply dropping these columns.

The Weibull AFT model trained on EHR data and tumour descriptors achieved the highest concordance index (C-Index) in the validation set (table 1). This model assumes the hazard probability to be a function of patient features x_i and time t parametrized by β_i and ρ ,

$$H(x_i, t; \beta_i, \rho) \propto - \left(\frac{t}{\exp(\beta_0 + \sum_i \beta_i x_i)} \right)^\rho.$$

The log-rank test revealed that tumour descriptors like mean PET intensity and number of tumours are significant for the predictions (table 2). The parameters ρ and β_0 determine the shape of the Weibull distribution and were fit to -3.07 and 34.24 for the model trained only on EHR, and -0.21 and 10.25 for the one trained on both EHR and tumour descriptors. We used the Lifelines package [24] implementation of Cox PH and Weibull AFT with their default parameters.

Table 1. Survival calibration results in the validation set in terms of concordance index. The Cox PH and the Weibull AFT models were fitted only on EHR data and both EHR and tumour descriptors. It can be seen that Weibull AFT outperforms Cox PH and that including tumour descriptors improves the results in both cases. Best figures are in bold.

| | Cox PH | Weibull AFT |
|--------------------------|---------|----------------|
| EHR | 0.58589 | 0.60996 |
| EHR + tumour descriptors | 0.60977 | 0.63014 |

Table 2. Log-rank test for the Weibull AFT model fit. The input features with lowest p-value are the most representative for the survival prediction task. Best figures in bold.

| Input feature (x_i) | Only EHR | | EHR + tumour descriptors | |
|-------------------------|---------------------------|-------------------------------|---------------------------|-------------------------------|
| | Coefficient (β_i) | p-value (\downarrow) | Coefficient (β_i) | p-value(\downarrow) |
| Age | -0.15 | 0.35 | -0.13 | 0.23 |
| Alcohol | 0.36 | 0.10 | 0.44 | 0.07 |
| Chemotherapy | -0.14 | 0.47 | -0.15 | 0.82 |
| Gender | -0.50 | 0.01 | -0.28 | 0.06 |
| HPV status (0=-, 1=+) | 0.37 | 0.08 | 0.31 | 0.12 |
| Performance status | -0.87 | \ll 0.005 | -0.69 | \ll 0.005 |
| Surgery | 0.16 | 0.37 | 0.15 | 0.65 |
| Tobacco | 0.13 | 0.54 | 0.03 | 0.40 |
| Weight | 0.42 | 0.03 | 0.55 | 0.01 |
| Area bounding box | - | - | -0.28 | \ll 0.005 |
| Centroid x coordinate | - | - | -0.64 | 0.22 |
| Centroid y coordinate | - | - | -1.41 | 0.99 |
| Centroid z coordinate | - | - | 0.92 | 0.23 |
| Max CT intensity | - | - | -0.27 | 0.05 |
| Mean CT intensity | - | - | 0.54 | 0.02 |
| Mean PET intensity | - | - | -0.55 | \ll 0.005 |
| Number of tumours | - | - | -0.43 | \ll 0.005 |

Survival Model. The winning method of the previous edition of the challenge, named *Deep Fusion* [25], consisted of a CNN encoder that takes a fused PET-CT image as an input, and outputs a feature embedding that is concatenated with patient EHR data. The final layer is a multi layer perceptron (MLP) connected with the multi task logistic regression (MTLR) loss, which can model individual risk scores accurately [16–18]. It divides the target time into bins for which survival scores are predicted, imposing constraints to deal with uncensored and censored events.

Since *number of tumours*, n , was one of the most representative features in the calibration experiments, we hypothesize that, rather than one single image patch, processing n fused PET-CT patches with graph convolution networks may provide stronger prognosis signals. In the proposed *Multi-patch* model, 64-dimensional patch embeddings were first obtained with a CNN layer followed by batch normalization [19], ReLU activation and average pooling. Next, we built an unweighted fully connected graph of n nodes, one for each of the image patches.

We assigned concatenations of the CNN embeddings of the image patches and the tumour descriptors selected in the *calibration experiments* as node features, and applied two layers of graph convolution and ReLU activation to reason over the tumour graph. The improved graph attention network (GATv2) [11] was chosen for its availability to perform dynamic node attention.

Finally, an average pooling layer generates global graph vector embeddings, which are then concatenated with each patient’s EHR data. A Multi Layer Perceptron (MLP) produces the output logits, which are then used to compute the MTLR loss and the patient’s risk score. The full pipeline is shown in Fig. 5.

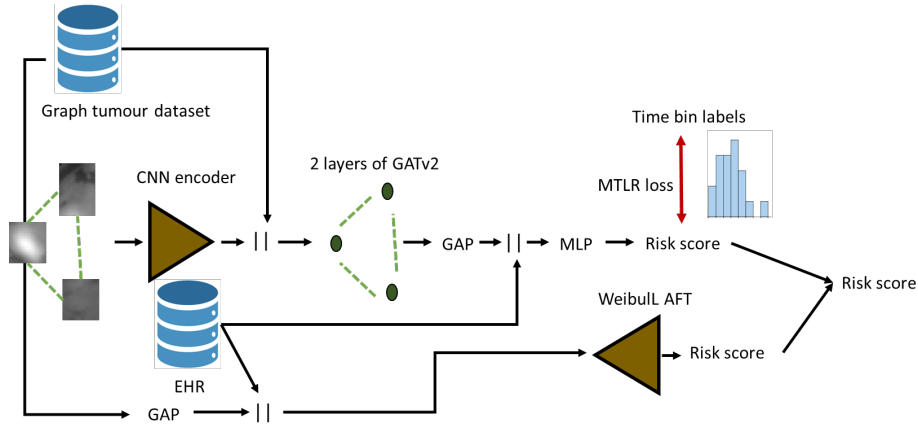


Fig. 5. Multi-patch network for survival prediction. The inputs are fused CT-PET image patches cropped at the segmented tumour centroids, tumour features, and patient EHR data. The output is the predicted risk score.

3 Experimental Set Up

Here we provide details of our experimental implementation, covering data splitting (§3.1), the preprocessing and augmentation techniques (§3.2), and the hardware and network hyperparameters (§3.3).

3.1 Data Splitting

For the segmentation task, the training data was divided into training and validation splits with 445 and 79 cases respectively for the segmentation task. To ensure a balanced representation of multi-centre data, the cases from each centre were first randomly allocated to *per-centre* subsplits, and then aggregated to form the final split.

Since the survival dataset is a subset of the segmentation one, we opted to define the survival splits as subsets of the segmentation partitions, resulting in 414 training and 74 validation cases. In this manner the inferred risk score is determined by the segmentation inference. We confirmed (Fig. 6) the absence of important distribution shifts between the training and validation survivals times and proportion of censoring cases.

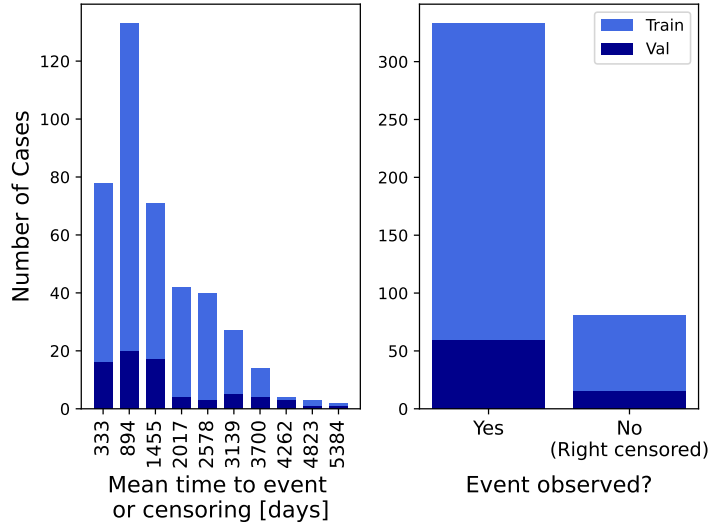


Fig. 6. Histograms of survival times and censoring status of the training and validation cases. It can be seen that both distributions are very similar, peaking at the same mean. Best seen in colour.

3.2 Data Preprocessing and Augmentations

Because global context is more important than detail for localization, we resampled the images to low resolution, e.g. $(7, 7, 7)$ mm voxel spacing and clipped the CT values to the interval $(-1024, 1024)$. Instead, for the segmentation and survival networks, where detail or texture is more important, we resampled the inputs to 1 mm isotropic voxel size and windowed the CT values to $(-200, 200)$ to enhance soft tissues.

In all cases, the images were normalized by subtracting the mean and dividing by the standard deviation. The input of the localization and segmentation networks are pairs of CT and PET images, whereas for the survival network, the PET and CT image are fused by averaging.

For the localization model, the following random augmentations (with probability p) were applied the training samples: random intensity shifts in the range $(-0.5, 0.5)$ ($p = 0.5$), random scale shifts with in the range $(-1, 1)$ for the PET and $(-0.25, 0.25)$ for the CT ($p = 0.5$), and random Gaussian Noise with mean 0 and standard deviation 0.1 ($p = 0.1$). The same augmentations reported in [10] were used to train the binary segmentation network. No augmentations were used for the survival prediction models.

3.3 Implementation Details

All models were run in a 32 GB NVidia Tesla V100, although they had different memory footprints and training times. The localization model was the lightest,

whereas the binary segmentation model was the heaviest and with longer training time. All models were implemented using PyTorch [27], PyTorch Lighting [28] and PyTorch Geometric [29]. Table 3 summarizes the different hyperparameters and other training details for these three models.

Table 3. Implementation and training details of the different trained networks in this study.

| | Detection | Segmentation | Survival (Deep Fusion) | Survival (Multi-patch) |
|--------------------------|-------------------------|----------------------|---------------------------|---------------------------|
| Optimizer | Adam | SGD with momentum | Adam | Adam |
| Scheduler | Reduce LR on plateau | Poly LR | Multi step LR | Multi step LR |
| Initial learning rate | 0.001 | 0.001 | 0.016 | 0.016 |
| Loss function | Dice focal | Dice focal | MTLR | MTLR |
| Epoch | 100 | 100 | 100 | 100 |
| GPU RAM (GB) | 4 | 26 | 10 | 10 |
| Patch size | 64^3 | 128^3 | $80 \times 80 \times 50$ | 32^3 |
| Training time (hours) | 2 | 48 | 10 | 10 |
| Validation metric | Average precision | Average precision | Concordance index | Concordance index |

4 Results

Here we present our results, first for the segmentation task (§4.1), then for the survival prediction task (§4.2).

4.1 Segmentation Results

First, we assessed the binary segmentation and classification performance separately in the validation set. The binary segmentation network achieves a Dice score of 0.636, whereas for the tumour classification problem, the macro and micro F1 scores are 0.843 and 0.861 respectively.

Then, we obtained the semantic segmentation outputs from the binary segmentations and classifications results, and computed the aggregated Dice score.

All the intersections are divided by all the unions in the considered data split independently for each class, and then the mean of the two is computed.

Table 4 reports this metric in the validation and test sets. Although the results suggest the proposed model is not a strong segmentor, it provides suitable input for the survival task, which benefits from the features extracted from the segmented tumours.

Table 4. Results of the proposed segmentation method in the validation and test sets.

| Dataset Split | GTVp Dice | GTVn Dice | Aggregated Dice |
|---------------------------|-----------|-----------|-----------------|
| Validation Set (74 cases) | 0.68514 | 0.62648 | 0.65581 |
| Test Set (359 cases) | 0.59424 | 0.54988 | 0.57206 |

Finally, we qualitatively assessed the algorithm outputs by looking at the best, average and worst cases (Fig. 7). As a result of mistakes in the localization step, some of the predicted bounding box were slightly shifted from the actual RoI, which in turn resulted errors in the segmentation stage. For example, in *MDA-036* the predicted bounding box included a greater part of the brain, and the segmentation network detected a small region of brain as the tumour. Therefore, post-processing of the predicted bounding box could have improved the training stability and segmentation results.

4.2 Survival Prediction Results

Here we present our results for the survival prediction task. Table 5 reports the C-Index achieved by the proposed methods in the validation and test sets. For a fair comparison, we implemented and reported the results of *Deep Fusion*, the best performing neural network presented last year [25] for this task.

Table 5. Results of different methods in the validation set in terms of Concordance Index (C-Index). Best results in bold.

| Model | Validation Set (74 images) | Test Set (339 images) |
|----------------------------------|----------------------------|-----------------------|
| Weibull AFT | 0.63014 | 0.64086 |
| Deep Fusion [25] | 0.60587 | 0.47923 |
| Deep Fusion [25] + Weibull AFT | 0.72194 | 0.64081 |
| Multi-patch (ours) | 0.75000 | 0.39679 |
| Multi-patch + Weibull AFT (ours) | 0.70536 | 0.64013 |

The proposed Multi-patch method performs best in the validation set, followed by Deep Fusion. Nevertheless, both deep learning methods generalize

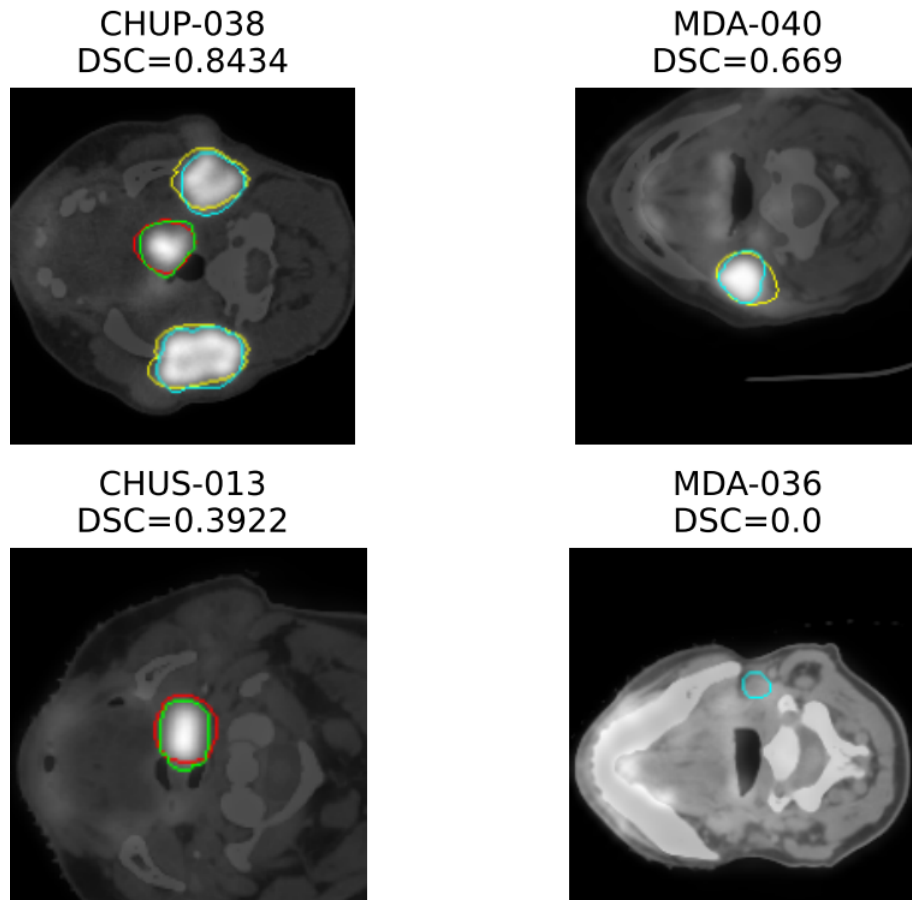


Fig. 7. Best, two average and worst segmentation outputs. The same preprocessing and averaging technique for the survival prediction is applied to the CT and PET images. Ground truth for GTV_p and GTV_n contours are in green and cyan, whereas predicted contours are in red and yellow, respectively.

poorly to the test set, with our method obtaining the worst metric. To try to mitigate the overfitting, we ensembled the outputs of the deep learning and the Weibull AFT methods by simple averaging. However, even under this setting, the Weibull AFT model alone was the best performing method in the test set. Little difference was observed between this model alone and ensembles of it and the deep learning algorithms.

5 Discussion & Conclusion

We presented a framework for tumour segmentation and prognosis prediction in head & neck cancer patients, which may have a positive impact in patient management and personalized healthcare. Nevertheless, generalization still poses a challenge to the adoption of a solution based on neural networks. This has resulted in worse performance of the segmentation model, and even more so the survival model in the unseen cases of the test set.

The good generalization of Weibull AFT may due to the fewer parameters of this model compared with their deep learning counterparts, greatly reducing the possibility of overfitting. Some possible ways to mitigate this include reducing the neural network capacity (number of parameters), and to use *n-fold* cross validation and regularization during training. On the other hand, the superior performance of Weibull AFT with respect to Cox PH can be attributed to the acceleration/deceleration effect of the input features on the hazard probability, rather than their time independence, as assumed by the Cox PH model.

Finally, we have incorporated tumour-instance information in the prediction via processing tumour descriptors and tumour centred image patches with the improved graph attention networks [11]. Explanation algorithms like the approximated Shapley values [31] could be combined with the proposed method to increase the interpretability of predictions, a matter of crucial importance in clinical practice. Similar approaches have been used for gene expression data [32] and histopathology images [33]. We leave the application of these methods to head & neck cancer as future work.

References

1. Sung, Hyuna, et al.: Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians* 71.3 : 209–249 (2021)
2. Johnson, D.E., Burtness, B., Leemans, C.R. et al. Head and neck squamous cell carcinoma. *Nat Rev Dis Primers* 6, 92 (2020)
3. Andrearczyk, Vincent, et al. "Overview of the HECKTOR challenge at MICCAI 2020: automatic head and neck tumor segmentation in PET/CT." *3D Head and Neck Tumor Segmentation in PET/CT Challenge*. Springer, Cham, 2020.
4. Oreiller V., et al., "Head and Neck Tumor Segmentation in PET/CT: The HECKTOR Challenge", *Medical Image Analysis*, 77:102336 (2022).

5. Andrearczyk, Vincent, et al. "Overview of the HECKTOR challenge at MICCAI 2021: automatic head and neck tumor segmentation and outcome prediction in PET/CT images." *3D Head and Neck Tumor Segmentation in PET/CT Challenge*. Springer, Cham, 2021.
6. Andrearczyk V, et al. "Overview of the HECKTOR Challenge at MICCAI 2022: Automatic Head and Neck Tumor Segmentation and Outcome Prediction in PET/CT", in: *Head and Neck Tumor Segmentation and Outcome Prediction, (2023)*
7. Ronneberger, Olaf et al.: "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.
8. Isensee, Fabian, et al. "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation." *Nature methods* 18.2 (2021): 203-211.
9. Iantsen A., Visvikis D., Hatt M. (2021) Squeeze-and-Excitation Normalization for Automated Delineation of Head and Neck Primary Tumors in Combined PET and CT Images. In: Andrearczyk V., Oreiller V., Depeursinge A. (eds) *Head and Neck Tumor Segmentation. HECKTOR 2020. Lecture Notes in Computer Science*, vol 12603. Springer, Cham. https://doi.org/10.1007/978-3-030-67194-5_4
10. Xie, J., Peng, Y. (2022). The Head and Neck Tumor Segmentation Based on 3D U-Net. In: Andrearczyk, V., Oreiller, V., Hatt, M., Depeursinge, A. (eds) *Head and Neck Tumor Segmentation and Outcome Prediction. HECKTOR 2021. Lecture Notes in Computer Science*, vol 13209. Springer, Cham. https://doi.org/10.1007/978-3-030-98253-9_8
11. Brody, S., Alon, U., & Yahav, E. (2021). How attentive are graph attention networks?. *arXiv preprint arXiv:2105.14491*.
12. Van der Walt, Stefan, et al. "scikit-image: image processing in Python." *PeerJ* 2 (2014): e453.
13. Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *the Journal of machine Learning research* 12 (2011): 2825-2830.
14. Kalbfleisch, J.D., Prentice, R.L.: *The Statistical Analysis of Failure Time Data*. Wiley, New York (1980)
15. Cox, D.R.: Regression models and life-tables. *J. Roy. Stat. Soc. Ser. B (Methodol.)* 34(2), 187–202 (1972). <https://doi.org/10.1111/j.2517-6161.1972.tb00899.x>
16. Yu, Chun-Nam, et al. "Learning patient-specific cancer survival distributions as a sequence of dependent regressors." *Advances in neural information processing systems* 24 (2011).
17. P. Jin, 'Using Survival Prediction Techniques to Learn Consumer-Specific Reservation Price Distributions', University of Alberta, Edmonton, AB, 2015.
18. S. Fotso, et al. 'Deep Neural Networks for Survival Analysis Based on a Multi-Task Framework', *arXiv:1801.05512*.
19. Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). PMLR.
20. Milletari, Fausto, et al. "V-net: Fully convolutional neural networks for volumetric medical image segmentation." *2016 fourth international conference on 3D vision (3DV)*. IEEE, 2016.
21. Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.
22. Lin, Tsung-Yi, et al. "Focal loss for dense object detection." *Proceedings of the IEEE ICCV*. 2017.

23. Sudre, C. et al. (2017) Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. DLMIA 2017
24. Davidson-Pilon, C. et al. "Lifelines: survival analysis in Python." *J. Open Source Softw.* 4(40), 1317 (2019). <https://doi.org/10.21105/joss.01317> (2019)
25. Saeed, N., Al Majzoub, R., Sobirov, I., Yaqub, M. (2022). An Ensemble Approach for Patient Prognosis of Head and Neck Tumor Using Multimodal Data. In: Andreczyk, V., Oreiller, V., Hatt, M., Depeursinge, A. (eds) *Head and Neck Tumor Segmentation and Outcome Prediction. HECKTOR 2021. Lecture Notes in Computer Science*, vol 13209. Springer, Cham.
26. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next-generation hyperparameter optimization framework. *CoRR* abs/1907.10902 (2019).
27. Paszke, A., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32 (2019).
28. Falcon, William, et al. "PyTorch Lightning",doi: 10.5281/zenodo.3828935 (2019).
29. Fey, M.,et al. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428*. (2019)
30. Fatima, S. S., et al. "A linear approximation method for the Shapley value". *Artificial Intelligence*, 172(14), 1673-1699.(2008)
31. Ancona, M., et al. "Explaining deep neural networks with a polynomial time algorithm for shapley value approximation." In *ICML* (pp. 272-281). PMLR. (2019, May)
32. Hayakawa, Jin, et al. "Pathway importance by graph convolutional network and Shapley additive explanations in gene expression phenotype of diffuse large B-cell lymphoma." *Plos one* 17.6 (2022): e0269570.
33. Bhattacharjee, Subrata, et al. "An Explainable Computer Vision in Histopathology: Techniques for Interpreting Black Box Model." *International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*. IEEE, 2022.