Brain Tumor Segmentation and Survival Prediction

Machine & Deep Learning Project Report

Vincenzo Colle
M.Sc. in Telecommunications Engineering
Department of Electrical and Information Engineering
Università degli Studi di Cassino e del Lazio Meridionale
vincenzo.colle@studentmail.unicas.it

Supervisors
Prof. Claudio Marrocco
Prof. Alessandro Bria

Abstract

Brain tumor segmentation and survival prediction are two clinically relevant and challenging tasks in medical image analysis. In this project, we utilize the BraTS2020 dataset, which provides multimodal 3D magnetic resonance imaging (MRI) scans with voxel-level tumor annotations and patient survival information. For the segmentation task, we adopt a 3D U-Net architecture as a baseline and introduce several architectural modifications to improve performance and training stability. For survival prediction, we initially investigate the use of the segmentation network's bottleneck as a source of compact patient-level features. However, principal component analysis reveals a severe collapse of the learned representation, rendering it unsuitable for classical machine learning models. To address this limitation, we employ an AutoEncoder to learn richer latent representations, enabling binary survival classification on a subset of patients. The results highlight both the strengths and limitations of the proposed approaches under computational constraints.

1 Introduction

Gliomas are among the most aggressive and heterogeneous brain tumors, requiring accurate delineation and prognosis estimation to support clinical decision-making. Magnetic resonance imaging (MRI) is the primary imaging modality used for diagnosis and follow-up, offering multiple contrasts that capture complementary anatomical and pathological information. Automated analysis of multimodal MRI data has therefore become a central research topic in medical image computing.
The Brain Tumor Segmentation (BraTS) challenges provide standardized datasets and evaluation protocols to advance research in tumor segmentation and survival prediction. In this project, we focus on the BraTS2020 dataset and address two main tasks: voxel-wise brain tumor segmentation and patient-level survival prediction. Our contributions are threefold:

1. We design and implement a modified 3D U-Net architecture for brain tumor segmentation that complements standard encoder-decoder skip connections with residual connections within each resolution stage and Squeeze-and-Excitation (SE) modules.

2. We analyze the suitability of bottleneck features learned by the segmentation network for survival prediction and demonstrate, through Principal Component Analysis (PCA), a pronounced *representation collapse* that limits their prognostic capacity.

3. We propose an AutoEncoder-based latent feature learning strategy that enables richer and more discriminative latent representations for survival prediction.

2 Dataset Overview

2.1 BraTS2020 Dataset Characteristics

The BraTS2020 dataset [1] consists of multimodal three-dimensional MRI scans acquired from patients diagnosed with gliomas. For each subject, four MRI modalities are provided: native (T1), post-contrast T1-weighted (T1ce), T2-weighted (T2) and Fluid-Attenuated Inversion Recovery (FLAIR). The scans are provided with standardized pre-processing: they are co-registered[1] to the same anatomical template, interpolated to an isotropic resolution of 1 mm^3 and skull-stripped to remove non-brain tissue. All modalities are supplied at a spatial resolution of $240 \times 240 \times 155$ voxels.

The dataset is divided into three subsets with distinct purposes. The training set comprises 369 subjects and includes full voxel-wise ground truth annotations, which are used to train supervised segmentation models. The validation set contains 125 subjects but does not provide segmentation masks; within the context of the BraTS challenge, predictions for these cases are submitted to an evaluation server for scoring. Although these subjects cannot be used for segmentation, due to the absence of labels, we leverage them in the training of the AutoEncoder. Finally, the test set consists of a private collection of subjects reserved for the final ranking of challenge participants and was not released publicly.

3 Brain Tumor Segmentation

3.1 Data Pipeline

3.1.1 Data Preprocessing

Given the multi-modal nature of the data (4 channels) and the limited GPU memory available for this project, training on the full $240 \times 240 \times 155$ volumes was not feasible. To address this, we perform a center crop to a resolution of $128 \times 128 \times 128$ voxels. While we acknowledge that such a cropping strategy may occasionally exclude relevant brain regions or tumor boundaries, it was a mandatory trade-off to enable multi-modal training within our memory constraints. Figure 1 illustrates an example of the cropping on a sample MRI volume, in T1 native modality. It is evident that the cropping operation mostly affects the background regions, but also impacts useful brain portions. Table 1 compares the labels distribution in the original MRI volumes versus the cropped volumes.

Label	Description	Original %	Cropped %
0	Air	83.78%	40.02%
	Healthy brain tissue	15.10%	55.42%
1	Necrotic and non-enhancing tumor core	0.25%	1.02%
2	Peritumoral edema	0.65%	2.64%
4	Enhancing tumor	0.22%	0.90%

Table 1: Comparison of average voxel-wise labels distribution in original volumes versus center-cropped volumes.

[1]Co-registered means that all MRI modalities are spatially aligned to the same anatomical reference frame, so that corresponding voxels represent the same physical location in the brain.

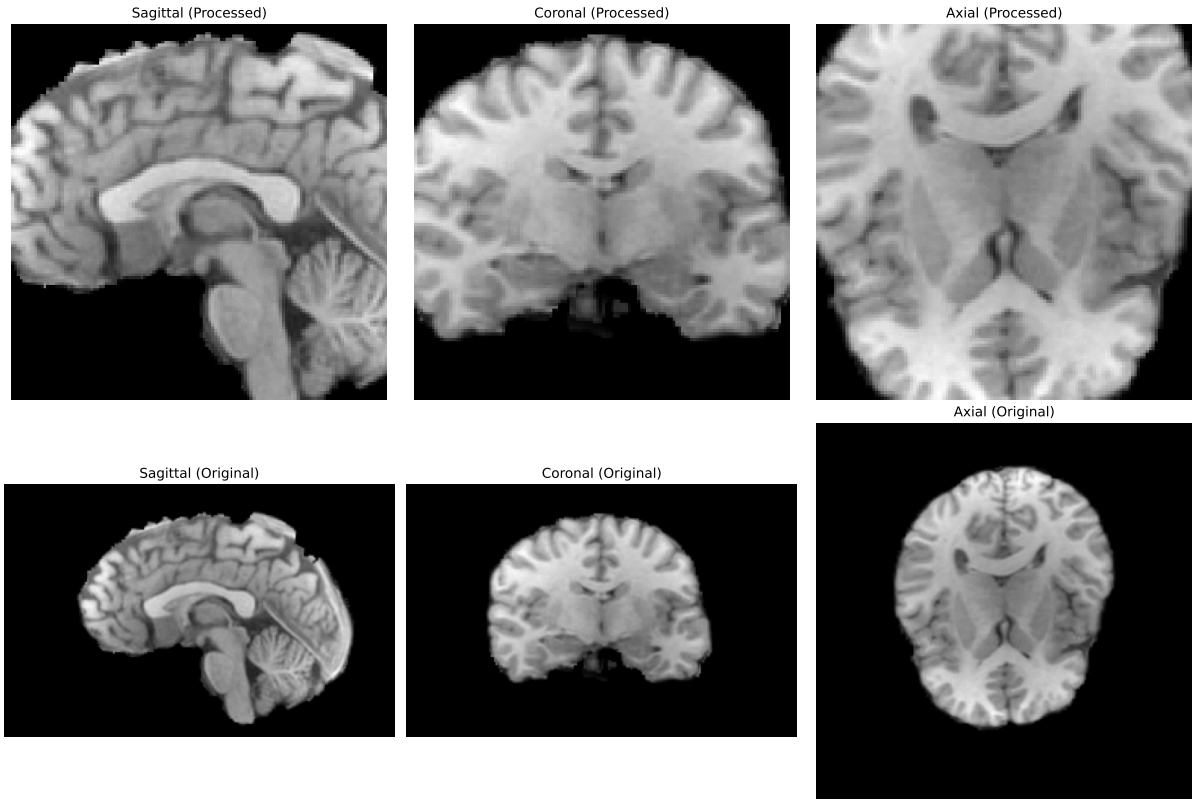


Figure 1: Example of center cropping applied to a sample MRI volume (T1 native modality). Tumor regions near the edges may be partially excluded.

To quantify how much of the tumor tissue is left out, on average, we compute the fraction of voxels for each tumor subregion before and after cropping. The results are reported in Table 2. Overall, more than 95% of voxels corresponding to necrotic/non-enhancing tumor core (label 1), peritumoral edema (label 2) and enhancing tumor (label 4) are preserved, indicating that the majority of relevant tumor tissue for segmentation is retained.

Label	**Original Avg.**	**Cropped Avg.**	**Lost (%)**	**Retained (%)**
1	22,118	21,440	3.01%	96.99%
2	57,727	55,403	4.04%	95.96%
4	19,697	18,840	4.39%	95.61%

Table 2: Quantitative analysis of tumor voxel retention after center cropping to $128 \times 128 \times 128$. The minimal loss ($< 5\%$) validates the cropping strategy.

To better distinguish healthy brain tissue from air, we introduce an additional label (label 5) representing healthy brain voxels. We evaluate this design choice by comparing models trained with and without the added label, to test whether learning normal anatomy improves tumor boundary discrimination. Finally, the 369 annotated subjects are divided into a training set and a validation set using an 80%-20% split, resulting in 295 subjects for training and 74 subjects for validation.

3.1.2 Data Augmentation.

Data augmentation is a critical strategy in medical image analysis to mitigate overfitting, particularly given the relatively small size of datasets like BraTS. For this project, we implemented

an on-the-fly augmentation pipeline, using the `TorchIO` library, which allows for consistent transformations across the four MRI modalities.

Theoretical Background. Data augmentation techniques in medical imaging can be broadly categorized into geometric and intensity-based transformations. Geometric transformations are designed to simulate variations in patient positioning and anatomical structure; these include rigid affine operations such as rotation, translation and scaling, as well as flipping and elastic deformations that model non-rigid tissue distorsions. Intensity-based transformations aim to replicate scanner-dependent artifacts and acquisition variability and typically involve random noise injection, Gaussian blurring, gamma-based contrast adjustment and simulation of MRI bias fields to account for magnetic field inhomogeneities.

Implementation. Unlike offline preprocessing, on-the-fly transformations introduce a computational overhead during data loading that must be balanced against training speed. While realistic data augmentations such as elastic deformations and bias field simulation are commonly employed in brain MRI segmentation, their computational cost was prohibitive in our experimental environment.
Through profiling, we determined that both transformations significantly bottlenecked the data loader. Specifically, we tested the `RandomBiasField` transform provided by `TorchIO`, estimating its impact on training time, per epoch, to be approximately:

$$\Delta t \approx p \cdot N \cdot t_{\text{op}} \approx 0.2 \cdot 295 \cdot 15 \text{ s} \approx 885 \text{ s } (\sim 15 \text{ min}),$$

where p is the probability of applying the transformation (0.2), N is the number of training subjects (295) and t_{op} is the average time per operation per subject (15 s). On the single-core Intel(R) Xeon(R) CPU @ 2.20 GHz used for training, this overhead translated to roughly 15 minutes of GPU idle time per epoch, which was deemed unacceptable. Consequently, bias field transformations were excluded and similar considerations led to the omission of elastic deformations. Offline augmentation was considered but ultimately rejected due to prohibitive storage requirements and the need to preserve training stochasticity. Therefore, based on these computational constraints, we restrict our data augmentation strategy to a small set of transformations that are both effective and computationally inexpensive. The final augmentation pipeline comprises the following operations:

Geometric: Random flipping along the Left-Right axis ($p = 0.5$) and random affine transformations ($p = \frac{1}{3}$), including isotropic scaling $s \in [0.6, 1.4]$, rotation $\theta \in [-15°, +15°]$ and translation $t \in [-5, +5]$ voxels.

Intensity: Random Gamma correction, where voxel intensities I are transformed as $I_{\text{out}} = I_{\text{in}}^{\gamma}$. The exponent is sampled from a log-uniform distribution such that $\log(\gamma) \in [-0.3, 0.3]$.

After augmenting, each modality is standardized by subtracting its mean intensity and dividing by its standard deviation (standard normalization). These statistics are computed exclusively on non-zero voxels, to avoid introducing bias from the dominant background region. Figure 2 shows an example of an augmented and normalized MRI volume.

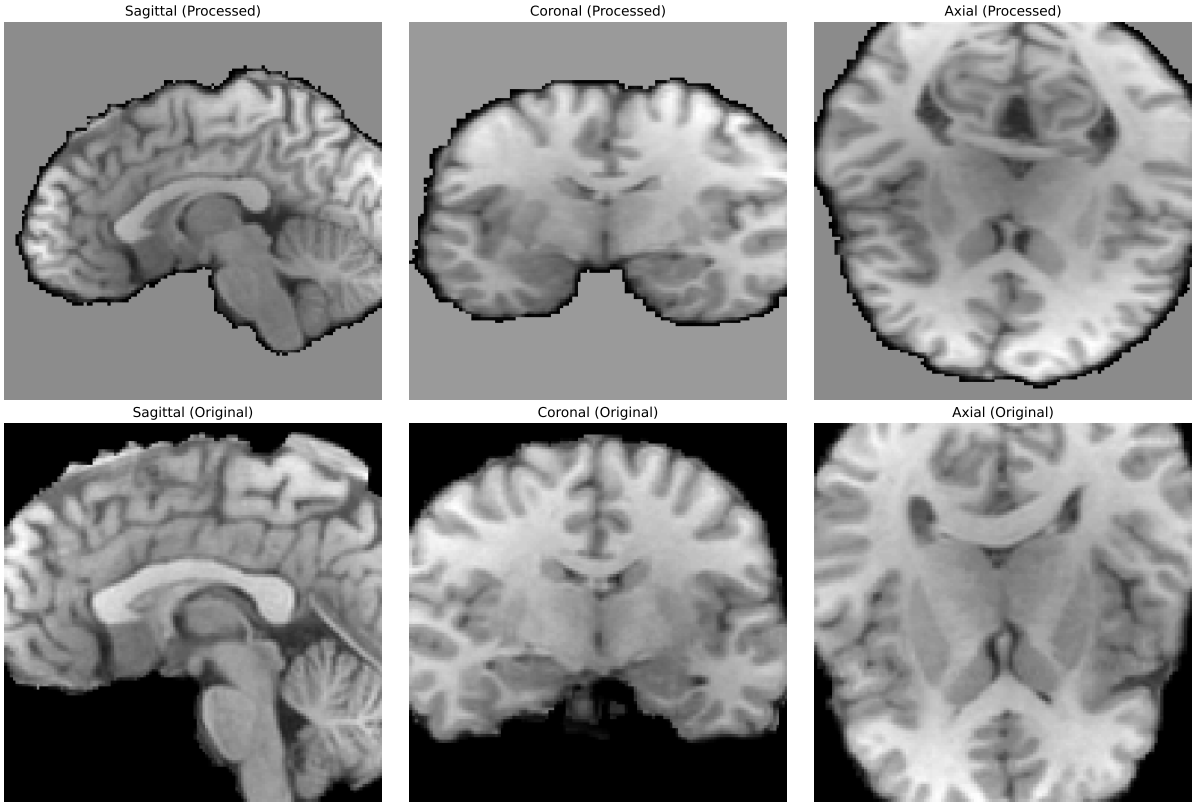


Figure 2: Example T1 native MRI volume before (top) and after (bottom) geometric and intensity augmentations. Note: The 'Processed' images display the same slice indices as the 'Original' images. Due to the affine transformations applied, the anatomical structures have shifted out of the original field of view, resulting in a different anatomical cross-section being displayed.

3.2 Proposed Methodology

3.2.1 Network Architecture.

For tumor segmentation, we adopt the 3D U-Net architecture proposed by Çiçek et al. [2] as our baseline. The network follows an encoder-decoder structure with skip connections that combine low-level spatial detail with high-level semantic information. Our implementation consists of five hierarchical encoding-decoding levels. The number of feature channels in the encoder doubles at each downsampling step (16, 32, 64, 128, 256), while the spatial resolution is halved. In the decoder, this process is reversed: the number of channels decreases symmetrically as the spatial resolution is progressively doubled. Each level of the network employs a block composed of two consecutive 3D convolutional layers, each followed by Group Normalization ($n_{\text{groups}} = 8$) and a ReLU activation. Convolutions are padded to preserve spatial dimensions. Group Normalization is preferred over Batch Normalization due to its improved stability when training with small batch sizes [3]. Table 3 summarizes the architectural specifications of the proposed model.

Residual Connections. In addition to the standard encoder–decoder skip connections of the original 3D U-Net, we incorporate intra-block residual connections. Here, we define a block as a single encoder or decoder stage, consisting of a sequence of convolutional operations applied at a fixed spatial resolution (i.e., the input feature map enters the block, undergoes multiple convolutions and exits the block before any resolution change). Within each block, we add a residual connection that links the block input directly to the block output, such that the

output of the final convolution is added element-wise to the original input of that block. These residual connections improve gradient flow and accelerate convergence, allowing the network to train substantially faster than architectures without such links, as reported in [4]. Implementing these residual connections requires the input and output of a block to have identical dimensions. However, in the original 3D U-Net, channel dimensionality changes occur within the second convolutional block, while spatial resolution changes are handled by separate downsampling or upsampling operations. This separation results in mismatched dimensions at the boundaries of encoder and decoder stages, preventing straightforward residual connections. To address this, we modify the architecture so that spatial resolution changes and channel scaling occur simultaneously. In the encoder, this is achieved using strided convolutions with a kernel size of 3, stride of 2 and padding of 1. Correspondingly, the decoder utilizes transposed convolutions for these transitions, employing a kernel size of 2 and a stride of 2. This design ensures that the feature map entering a block is already scaled to the required channel depth and spatial resolution, providing the necessary compatibility for element-wise addition.

Squeeze-and-Excitation Blocks. To further enhance the representational capacity of the proposed architecture, we integrate Squeeze-and-Excitation (SE) blocks [5].
SE blocks implement a channel-wise mechanism that models interdependencies between feature channels. By aggregating global spatial information through global average pooling, the SE block learns to recalibrate channel responses, emphasizing informative features while suppressing less relevant ones. This improves feature selectivity with negligible additional computational overhead.
We incorporate these blocks at two strategic locations:

1. Prior to the encoder-decoder skip connections: this allows the network to suppress redundant or useless feature maps and highlight more informative channels before they are concatenated with the upsampled decoder features.

2. Immediately after the each sequence of convolutional operations in the decoder pathway: by recalibrating the bottleneck and upsampled features, the network can better focus on relevant spatial and semantic information.

Formally, given an input feature map $X \in \mathbb{R}^{C\times H\times W\times D}$, the SE block performs two operations:

Squeeze. Global average pooling is applied across the spatial dimensions to produce a channel descriptor $z \in \mathbb{R}^C$:

$$z_c = \frac{1}{HWD}\sum_{i=1}^{H}\sum_{j=1}^{W}\sum_{k=1}^{D} X_{c,i,j,k}, \quad c = 1,\dots,C.$$

Excitation. The channel descriptor is passed through a two-layer fully connected network with a non-linearity to capture channel-wise dependencies and generate scaling factors $s \in \mathbb{R}^C$:

$$s = \sigma(W_2\,\delta(W_1 z)),$$

where $W_1 \in \mathbb{R}^{\frac{C}{r}\times C}$ and $W_2 \in \mathbb{R}^{C\times\frac{C}{r}}$ are learnable weights, r is the reduction ratio, δ is the ReLU activation and σ is the sigmoid function.
Finally, the input feature map is rescaled channel-wise:

$$\tilde{X}_c = s_c \cdot X_c, \quad c = 1,\dots,C,$$

effectively enhancing informative channels and suppressing less relevant ones.

3.2.2 Loss Function.

The Dice coefficient is a common metric for evaluating segmentation overlap between a predicted binary mask $P \in \{0,1\}^{H\times W\times D}$ and the corresponding ground truth $T \in \{0,1\}^{H\times W\times D}$:

$$\text{Dice}(P,T) = \frac{2\sum_i P^i T^i}{\sum_i P^i + \sum_i T^i},$$

where i indexes the voxels. While the Dice coefficient effectively measures spatial overlap, it is not directly suitable as a training objective, as it is computed on hard, discretized predictions obtained via a non-differentiable argmax operation.

To enable training, we resolve to the soft Dice loss, which generalizes the Dice coefficient to operate on the network's continuous probability outputs. For each class c, let $P_c \in [0,1]^{H\times W\times D}$ denote the predicted probability map and $T_c \in \{0,1\}^{H\times W\times D}$ the corresponding ground truth mask. The soft Dice loss for class c is defined as:

$$\mathcal{L}_{Dice,c} = 1 - \frac{2\sum_i P_c^i T_c^i + \epsilon}{\sum_i P_c^i + \sum_i T_c^i + \epsilon},$$

where ϵ is a small constant to avoid division by zero.

The final multi-class Dice loss used in training is computed by averaging the per-class losses over all tumor classes, excluding the background:

$$\mathcal{L}_{Dice} = \frac{1}{|C'|}\sum_{c\in C'} \mathcal{L}_{Dice,c},$$

where C' is the set of all classes except the background. Optionally, each class could be weighted differently to account for class imbalance.

Additionally, we conduct exploratory experiments combining the soft Dice loss with a weighted cross-entropy loss, as suggested in prior work, including the winning solution of the BraTS2020 challenge [6]. Different values of the weighting factor β were tested to control the contribution of the cross-entropy term in the total loss:

$$\mathcal{L}_{total} = \mathcal{L}_{Dice} + \beta\,\mathcal{L}_{CE}.$$

Class weights for the cross-entropy loss were set to be inversely proportional to the voxel-wise frequencies of each label in the training set, thereby mitigating the impact of class imbalance.

Table 3: Architectural specifications of the proposed 3D U-Net model.

Parameter	**Specification**
Input Channels	4 [T1, T1ce, T2, FLAIR]
Input Resolution	$128 \times 128 \times 128$ (Center-crop)
Channels Progression	[16, 32, 64, 128, 256]
Normalization	Group Normalization ($n_{groups} = 8$)
Activation	ReLU
SE Blocks Reduction	$r = 8$
Optimizer	AdamW (Weight Decay 1×10^{-3})
Learning Rate	1×10^{-4} (Cosine Annealing $\rightarrow 5 \times 10^{-5}$)
Loss Function	Soft Dice Loss
Batch Size (Effective)	4 (Batch Size: 2, Grad. accum. steps: 2)
Hardware	1× NVIDIA Tesla T4

3.3 Experiments and Results

3.3.1 Implementation Details.

Our model is trained on the BraTS2020 dataset for 30 epochs using a batch size of 2 with gradient accumulation over 2 steps on an NVIDIA Tesla T4 GPU.
Training was performed using the AdamW optimizer [7] with a weight decay of 1×10^{-3}, which slightly penalizes large weights to reduce overfitting. We employ a cosine annealing learning-rate scheduler, which gradually decreases the learning rate from the initial value of 1×10^{-4} to a final value of 5×10^{-5} over the course of training. We also experimented with a learning rate warmup at the beginning of training, but observed no significant improvement; thus, only cosine annealing was used in the final experiments. This scheduling strategy allows stable convergence while reducing the learning rate towards the end of training for fine-grained optimization.

3.3.2 Evaluation Metrics.

Segmentation performance is assessed using the Dice coefficient computed for each individual tumor label (necrotic/non-enhancing core, edema, enhancing tumor).
Additionally, the BraTS challenge employs specific composite regions as its primary evaluation metrics, namely:

1. **Whole Tumor (WT):** encompasses all tumor-related voxels, including edema and the tumor core.
2. **Tumor Core (TC):** includes the necrotic and non-enhancing core as well as enhancing tumor regions.
3. **Active Tumor (ET):** corresponds to the enhancing tumor portion only.

For each sample, predictions are converted to discrete labels using an argmax over the network's output probabilities. The Dice coefficient is computed for WT, TC and ET for each sample and then averaged.
To track overall performance during training, we also maintain a composite metric defined as the average Dice over the three regions:

$$\mathcal{D}_{\text{BraTS}} = \frac{\mathcal{D}_{\text{WT}} + \mathcal{D}_{\text{TC}} + \mathcal{D}_{\text{ET}}}{3}.$$

Checkpoints are stored based on improvements in this composite metric. Finally, to provide an additional measure of segmentation overlap, we report the Jaccard Index, also known as Intersection over Union (IoU). It is defined as the intersection of the prediction P and ground truth T divided by their union:

$$J(P,T) = \frac{|P \cap T|}{|P \cup T|} = \frac{\sum_i P_i T_i}{\sum_i P_i + \sum_i T_i - \sum_i P_i T_i} \tag{1}$$

3.3.3 Results.

We evaluate the overall segmentation performance using the composite Dice score. Table 4 reports results obtained using only the soft Dice loss for all model variants, while Table 5 presents results for the combined loss functions evaluated on the Dice-only best-performing architecture. In the tables, "baseline" refers to our proposed architecture trained using only tumor labels. In addition, we evaluate two architectural variants: (i) the inclusion of an explicit healthy brain label (Label 5) to improve foreground-background separation and (ii) a deeper network configuration with an additional U-Net level (320 channels at the bottleneck).

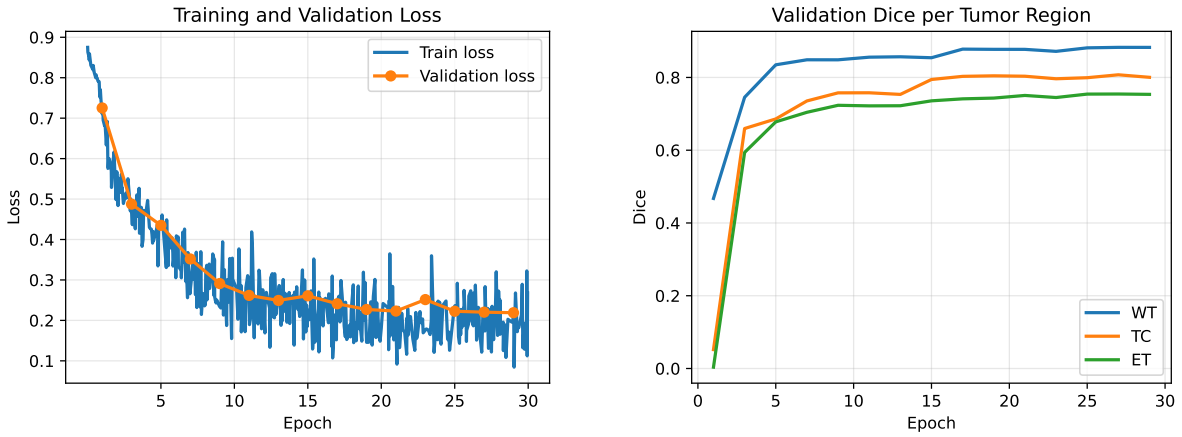
Table 4: Segmentation performance using soft Dice loss only. We report both Dice scores and Jaccard Index (IoU) for Whole Tumor (WT), Tumor Core (TC) and Enhancing Tumor (ET).

Configuration	**WT Dice** (%)	**WT IoU** (%)	**TC Dice** (%)	**TC IoU** (%)	**ET Dice** (%)	**ET IoU** (%)	**Overall** $\mathcal{D}_{BraTS}$ (%)
Baseline (BG ignored)	88.10	78.73	80.01	66.68	74.03	58.77	80.71
+ Healthy label (L5)	**88.27**	**79.00**	**80.73**	**67.69**	75.44	60.57	**81.48**
+ Deeper Arch.	88.00	78.57	79.25	65.63	**75.87**	**61.12**	81.04

Table 5: Segmentation performance using combined Dice and weighted Cross-Entropy loss. Jaccard Index (IoU) values are derived from the Dice scores.

Cross Entropy Loss (β)	**WT Dice** (%)	**WT IoU** (%)	**TC Dice** (%)	**TC IoU** (%)	**ET Dice** (%)	**ET IoU** (%)	**Overall** $\mathcal{D}_{BraTS}$ (%)
1.0	Failed to converge						-
2/3	Failed to converge						-
0.5	82.26	69.87	78.72	64.91	73.64	58.28	78.34
1/3	85.21	74.23	78.82	65.04	74.38	59.21	79.47

$\beta = 1.0$ and $\beta = \frac{2}{3}$ configurations failed to converge, as the cross-entropy term dominated the total loss and prevented meaningful improvements in Dice scores.
Figure 3 illustrates the training dynamics of our best-performing segmentation model, including loss curves and validation Dice scores for all three tumor regions. Figure 4 presents representative qualitative segmentation results obtained with the same configuration.



(a) Training and validation loss.

(b) Validation Dice for WT, TC and ET.

Figure 3: Training dynamics of the best-performing segmentation model.

3.3.4 Discussion.

The results in Table 4 demonstrate that soft Dice loss alone provides the strongest overall performance across all tumor regions. Introducing an explicit healthy brain label (Label 5) consistently improves segmentation accuracy, yielding the highest composite Dice score of 81.48%, after training for 30 epochs. This improvement suggests that explicitly modeling healthy brain tissue helps the network better distinguish tumor boundaries and reduces ambiguity between tumor and non-tumor regions. On the other hand, increasing the network depth beyond this configuration does not result in further improvements.
Table 5 shows that combining Dice loss with weighted cross-entropy is highly sensitive to the weighting factor β. Large values of β cause the cross-entropy term to dominate the optimization process, leading to unstable training and preventing meaningful improvements in Dice scores.

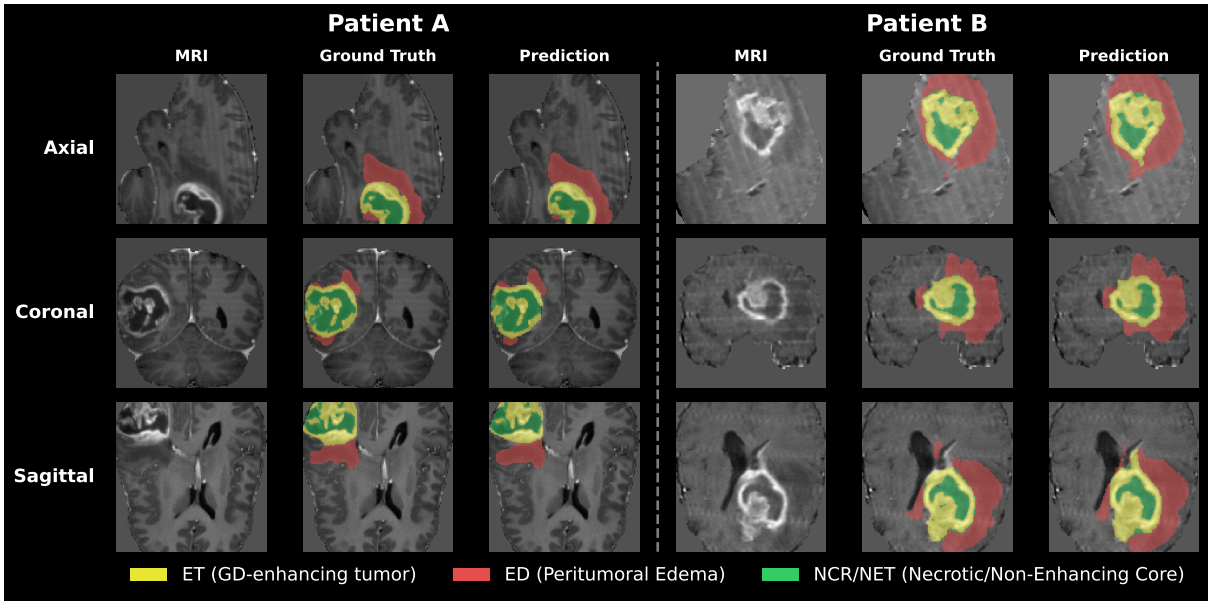


Figure 4: Comparison of two subjects across axial, coronal and sagittal views. Background: T1-weighted contrast-enhanced (T1ce) MRI. Columns show the input MRI, Ground Truth and Prediction. Colors: Green (Necrotic & Non-Enhancing Core), Red (Peritumoral Edema) and Yellow (Enhancing Tumor).

As β is reduced, performance gradually approaches that of Dice-only training, with the Dice component increasingly driving optimization. However, none of the combined-loss configurations outperformed the Dice-only baseline.
For context, the BraTS2020 challenge winner (nnU-Net [6]) achieved Dice scores of 88.95%, 85.06% and 82.03% for WT, TC and ET. Our best model approaches these values despite using only 80% of the training data, a simplified architecture and cropped inputs.

3.3.5 Limitations and Future Directions.

The primary methodological limitation of this work is the absence of ablation studies to isolate the contribution of individual architectural modifications. While we introduce residual connections and SE blocks and observe satisfactory performance, we cannot quantify how much each component contributes independently. Similarly, the inclusion of an explicit healthy brain label (Label 5) improved results, but this was tested only on a single architecture. Without broader validation, we cannot determine if this design choice is genuinely beneficial or simply happened to work well in this specific setup. Given the limited GPU time available for this project, conducting systematic experiments across multiple architectural variants was not feasible. Moreover, the most direct path to improved performance would be training at full $240 \times 240 \times 155$ resolution with comprehensive augmentation including elastic deformations and bias field corrections.
With better computational resources, future work should prioritize both ablation studies and full-resolution training with comprehensive augmentation: the former to understand which design choices truly matter and the latter to maximize absolute segmentation performance.

3.4 Principal Component Analysis

As a final experiment, we investigate the structure of the embedding space by applying Principal Component Analysis (PCA) to the features extracted from the segmentation network's bottleneck, right before the first upsampling operation in the decoder pathway. Specifically, for

each patient, the bottleneck output is a tensor

$$B \in \mathbb{R}^{C \times H \times W \times D},$$

where $C = 256$ and $H = W = D = 8$. To obtain a compact patient-level descriptor, we apply global average pooling across the spatial dimensions, yielding a 256-dimensional feature vector. Our analysis reveals an highly anisotropic embedding space: approximately 98% of the total variance is explained by a single principal component (PC1), with the second component capturing $\approx 1.5\%$.

Geometrically, the patient representations do not form a distinct 256-dimensional cloud but are strongly concentrated along a single direction. Additionally, the embeddings exhibit a large mean offset from the origin (mean norm ≈ 18.4 compared to an average spread of 1.6), suggesting the presence of a strong shared component or global bias common to all samples.

This behavior likely arises because the 3U-Net can rely on skip connections, leaving the bottleneck to encode little to no information. We tried training standard machine learning models on these features for our survival task, but no meaningful survival prediction could be achieved. Testing different bottleneck features extraction locations (e.g., before the bottleneck's residual) yielded similar results. Figure 5 shows the scree plot of the first 50 principal components, highlighting the sharp drop-off in explained variance.

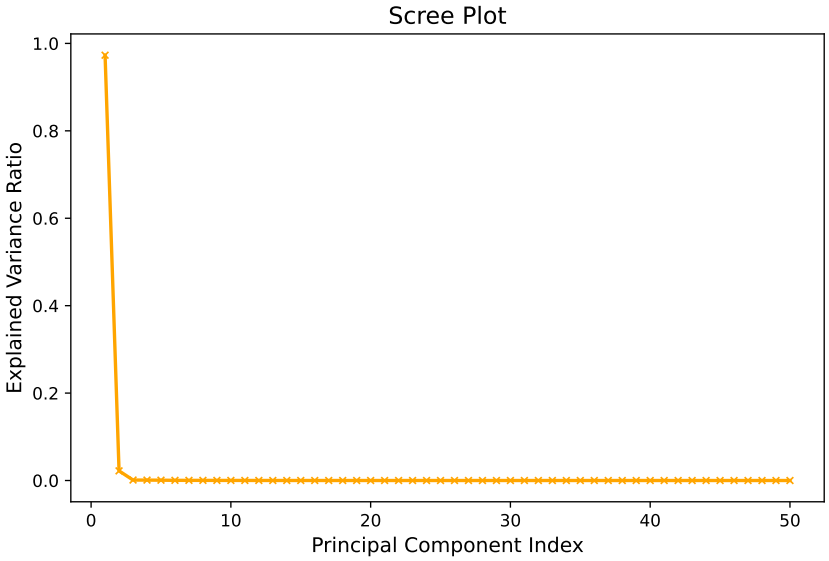


Figure 5: Scree plot of the first 50 bottleneck features. The sharp "elbow" at index 1 and the flat tail indicate that the effective dimensionality of the representation is close to 1, confirming severe dimensional collapse.

4 Survival Prediction via Latent Representation Learning

The severe representation collapse observed in the segmentation network bottleneck (Section 3.4) necessitated an alternative feature learning strategy. We hypothesize that this collapse is intrinsic to the U-Net architecture: the presence of encoder-to-decoder skip connections allows the decoder to retrieve high-frequency spatial details directly from early encoder layers, effectively bypassing the deepest part of the network. Consequently, the bottleneck is under-utilized and fails to capture a semantic summary of the input. To address this limitation, we transitioned from the segmentation network to a dedicated AutoEncoder model. The underlying hypothesis is that an unsupervised reconstruction objective, paired with the removal of skip

connections, will encourage the network to encode a richer, more distributed representation of anatomical and pathological patterns relevant to patient prognosis.
To this end, we adopt the AutoEncoderKL component from the framework presented in [8]. By employing the compression stage of this architecture, we project high-dimensional scans into a compact latent space without the computational overhead of the full diffusion model. Crucially, this architecture imposes a Kullback-Leibler (KL) divergence penalty on the encoder's posterior distribution, regularizing the predicted latent means and variances toward a shared standard normal prior. Without this constraint, the encoder could map inputs to arbitrary isolated latent points, optimized solely for reconstruction. The KL term discourages such degenerate deterministic encodings, encouraging a smooth and structured latent space that captures common factors of variation and is more compatible with the decision boundaries of classical machine learning models.

4.1 Data Pipeline

4.1.1 Data Preprocessing.

The data pipeline for the AutoEncoder was tailored specifically to the requirements of survival prediction and therefore differs from the preprocessing strategy used for segmentation. While segmentation tasks require the preservation of high-frequency details for accurate boundary delineation, necessitating high-resolution 128^3 crops, the AutoEncoder is designed to learn a compact and semantically meaningful representation of the patient's anatomy.
Accordingly, we prioritized a complete field-of-view over voxel-level resolution. This design choice is based on the hypothesis that macroscopic morphological patterns, such as tumor location and mass effect, are more prognostically informative for survival analysis than fine-grained texture. To incorporate this global anatomical context within the computational constraints of a single NVIDIA Tesla T4 GPU, volumes were resampled to $1.6 \times 1.6 \times 1.3, \text{mm}^3$ using B-spline interpolation, yielding an intermediate size of $150 \times 150 \times 119$, and subsequently cropped to a spatial resolution of $112 \times 112 \times 96$. This trade-off enabled preservation of the full brain context while accommodating the substantial memory overhead introduced by the AutoEncoder and PatchGAN discriminator. For similar computational reasons, training was restricted to the T1ce (post-contrast T1-weighted) modality rather than the full multi-modal input.
Exploiting the unsupervised nature of the AutoEncoder, we expanded the training set beyond the labeled data required for segmentation. Specifically, all available subjects were used, combining 369 annotated training cases with 125 unlabeled validation cases, resulting in a total of 494 volumes. Intensity normalization was performed using min-max scaling, computed over the 0th to 99.5th percentile range to reduce the influence of outliers.

4.2 Proposed Methodology

4.2.1 Network Architecture.

AutoEncoder. The AutoEncoderKL follows a hierarchical encoder-decoder architecture with the following specifications:

Encoder: The encoder consists of four downsampling levels with channel dimensions $(32, 64, 128, 128)$. Each level contains two residual blocks with Group Normalization ($n_{\text{groups}} = 8$) and ReLU activations. Spatial downsampling is performed via strided convolutions with a factor of 2 along each dimension, progressively reducing the input resolution from $112 \times 112 \times 96$ to $14 \times 14 \times 12$ at the bottleneck.

Latent: The bottleneck produces a 3-channel latent representation $z \in \mathbb{R}^{3\times14\times14\times12}$. The encoder outputs both a mean $\mu \in \mathbb{R}^{3\times14\times14\times12}$ and log-variance $\log\sigma^2 \in \mathbb{R}^{3\times14\times14\times12}$, from

which latent samples are drawn via the reparameterization trick:

$$z = \mu + \sigma \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I),$$

where $\odot$ denotes element-wise multiplication.

Decoder: The decoder mirrors the encoder structure, using transposed convolutions for upsampling and reconstructing the input volume at the original resolution.

Discriminator. The discriminator is a convolutional neural network following the PatchGAN design [9]. In our implementation, it consists of three convolutional layers with 32 channels, kernel size 4, and stride 2, using LeakyReLU activations ($\alpha = 0.2$). Instead of producing a single scalar output, it outputs a spatial map of real/fake predictions, enabling it to assess local image realism. Operating on patches of the input allows the discriminator to focus on fine-grained details such as textures and edges, which provides more informative gradients to the generator and improves high-resolution image synthesis.

4.2.2 Loss Function.

The total training objective combines four complementary loss terms:

Reconstruction Loss. We use L1 (mean absolute error) loss to measure voxel-wise fidelity between the input x and reconstruction $\hat{x}$:

$$\mathcal{L}_{\text{recon}} = \frac{1}{N} \sum_{i=1}^{N} |x_i - \hat{x}_i|,$$

where N is the total number of voxels. L1 loss is often favored over L2 for its robustness to outliers; however, relying solely on either loss typically yields blurry reconstructions.

KL Divergence Loss. To regularize the latent space and encourage a smooth, continuous distribution, we minimize the KL divergence between the learned posterior $q(z|x)$ and a standard Gaussian prior $p(z) = \mathcal{N}(0, I)$:

$$\mathcal{L}_{\text{KL}} = \frac{1}{2} \sum_{c,h,w,d} \left(\mu_{c,h,w,d}^2 + \sigma_{c,h,w,d}^2 - \log(\sigma_{c,h,w,d}^2) - 1 \right).$$

This term prevents the encoder from learning arbitrarily complex and overfitted representations.

Perceptual Loss. To improve the semantic quality of reconstructions beyond pixel-level similarity, we incorporate a perceptual loss [10, 11] using a pretrained SqueezeNet [12] feature extractor. By comparing images in feature space rather than pixel space, perceptual loss encourages the model to preserve high-level visual characteristics, such as textures, edges, and anatomical structures, that align with how humans judge image quality. To reduce memory usage when processing 3D volumes, we use a `fake_3d_ratio` of 0.25, meaning only 25% of slices from each volume are passed through the 2D feature extractor, and their features are aggregated to approximate the full 3D perceptual information. This allows the model to preserve semantic consistency efficiently while keeping training computationally feasible.

Adversarial Loss. To further enhance reconstruction realism, we train a PatchGAN discriminator D that evaluates local image patches rather than entire volumes, allowing it to focus on high-frequency textural details. The discriminator learns to distinguish between real medical images x and reconstructions $\hat{x}$ produced by the autoencoder. This creates an adversarial training dynamic: the generator (autoencoder) is trained to produce reconstructions that fool the discriminator:

$$\mathcal{L}_{\text{adv}}^{G} = \mathbb{E}_{z\sim q(z|x)}\left[(D(\hat{x}) - 1)^2\right],$$

while the discriminator is trained to correctly classify real images as 1 and reconstructions as 0:

$$\mathcal{L}_{\text{adv}}^{D} = \mathbb{E}_{x\sim p_{\text{data}}}\left[(D(x) - 1)^2\right] + \mathbb{E}_{z\sim q(z|x)}\left[D(\hat{x})^2\right].$$

We use a least-squares GAN objective, which has been shown to provide more stable training and higher quality gradients than the original binary cross-entropy formulation [13].
The final weighted loss for the generator is:

$$\mathcal{L}_G = \mathcal{L}_{\text{recon}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{perc}}\mathcal{L}_{\text{perc}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}^{G},$$

where $\lambda_{\text{KL}} = 1 \times 10^{-6}$, $\lambda_{\text{perc}} = 0.001$ and $\lambda_{\text{adv}} = 0.01$.

4.2.3 Implementation Details.

Training is performed for 100 epochs using the Adam optimizer with separate learning rates for the generator and discriminator: $\eta_G = 1 \times 10^{-4}$ and $\eta_D = 5 \times 10^{-4}$, respectively. The higher learning rate for the discriminator accelerates the adversarial training process and can help reduce the risk of mode collapse by keeping the discriminator sufficiently strong relative to the generator. We employ gradient accumulation over 8 steps to simulate a larger effective batch size while maintaining GPU memory constraints (batch size = 1, effective batch size = 8).
The 494 available volumes are divided into 369 training and 125 validation volumes, following the original dataset split ($\approx$ 75-25% split). The AutoEncoder was trained on a single NVIDIA Tesla T4 GPU, with validation loss monitored every 5 epochs. Additional architectural details are summarized in Table 6. Training progress is shown in Figure 6, where the left panel depicts the reconstruction loss on the training set and the right panel shows the corresponding validation loss. As seen, the validation loss reached its minimum around epoch 85, with $\mathcal{L}_{\text{recon}} \approx 0.025$.

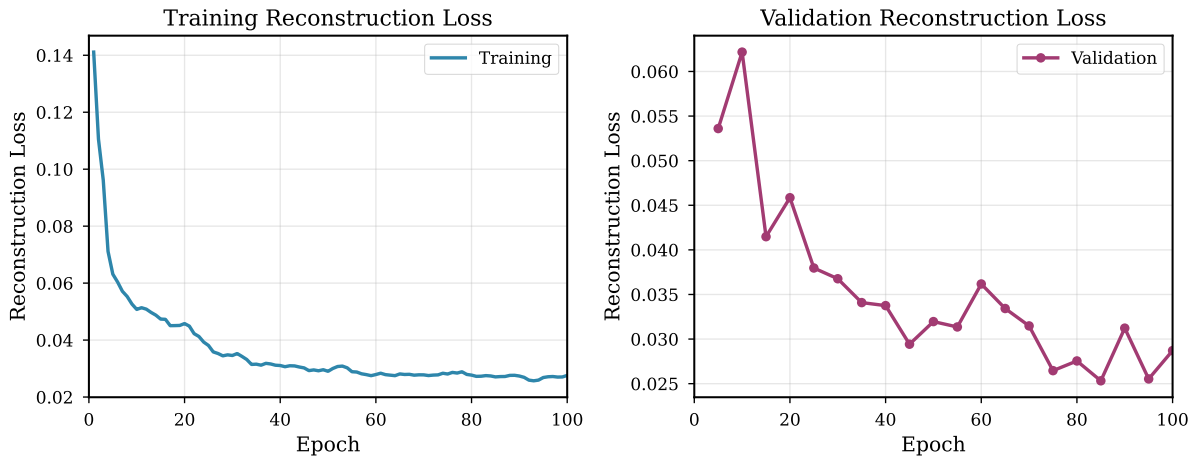


Figure 6: Training (left) and validation (right) reconstruction loss curves.

Table 6: Architectural specifications of the AutoEncoderKL and PatchGAN Discriminator.

Parameter	**Specification**
Generator (AutoEncoderKL)	
Input Modality	1× T1ce (Post-contrast)
Input Resolution	$112 \times 112 \times 96$
Latent Size	$3 \times 14 \times 14 \times 12$
Encoder Channels	[32, 64, 128, 128]
Residual Blocks per Level	1
Normalization	Group Normalization ($n_{groups} = 8$)
Pretrained init.	None[2]
Discriminator (PatchGAN)	
Layers	3
Channels	32
Kernel Size	$4 \times 4 \times 4$
Training Hyperparameters	
Weights	$\lambda_{rec} = 1, \lambda_{adv} = 0.01, \lambda_{KL} = 1e^{-6}, \lambda_{perc} = 0.001$
Optimizer	Adam
LR (Generator / Disc.)	1×10^{-4} / 5×10^{-4}
Batch Size (Effective)	8 (Batch Size: 1, Grad. accum. steps: 8)
Hardware	1× NVIDIA Tesla T4

4.2.4 Qualitative Results.

To assess the fidelity of the learned latent representations, we visually examine the AutoEncoder's reconstructions on validation subjects. Figure 7 displays the input T1ce volumes, the network's reconstruction and the voxel-wise absolute difference maps for two representative patients across axial, coronal and sagittal views.
The model successfully encodes global anatomical structures, preserving overall brain morphology, ventricular spaces and tumor bulk regions. The difference maps reveal that reconstruction errors are primarily concentrated at high-frequency anatomical details such as cortical sulci, tissue interfaces and edge regions, where intensity gradients are steepest. Seemingly, the bottleneck representations abstract away fine-grained textural information while retaining macrostructural and morphological characteristics, a compression that we expect to negatively influence the discriminative capacity of these features for downstream survival prediction tasks.

4.3 Survival Prediction Task

4.3.1 Problem Formulation.

Among the 494 subjects used for AutoEncoder training, survival information is available for 235 patients. We formulate survival prediction as a binary classification task by splitting patients at the median survival time of 370 days, resulting in two balanced groups: short survival ($<$ 370 days; 118 patients) and long survival ($\geq$ 370 days; 117 patients). The median split ensures balanced class sizes and avoids the severe imbalance that would arise from fixed time thresholds. This simplified formulation provides a baseline for evaluating the discriminative capacity of the learned latent representations.

[2]Ideally, we would have initialized from the UK Biobank pre-trained AutoEncoder by [14] but memory-driven architectural changes caused an incompatibility.

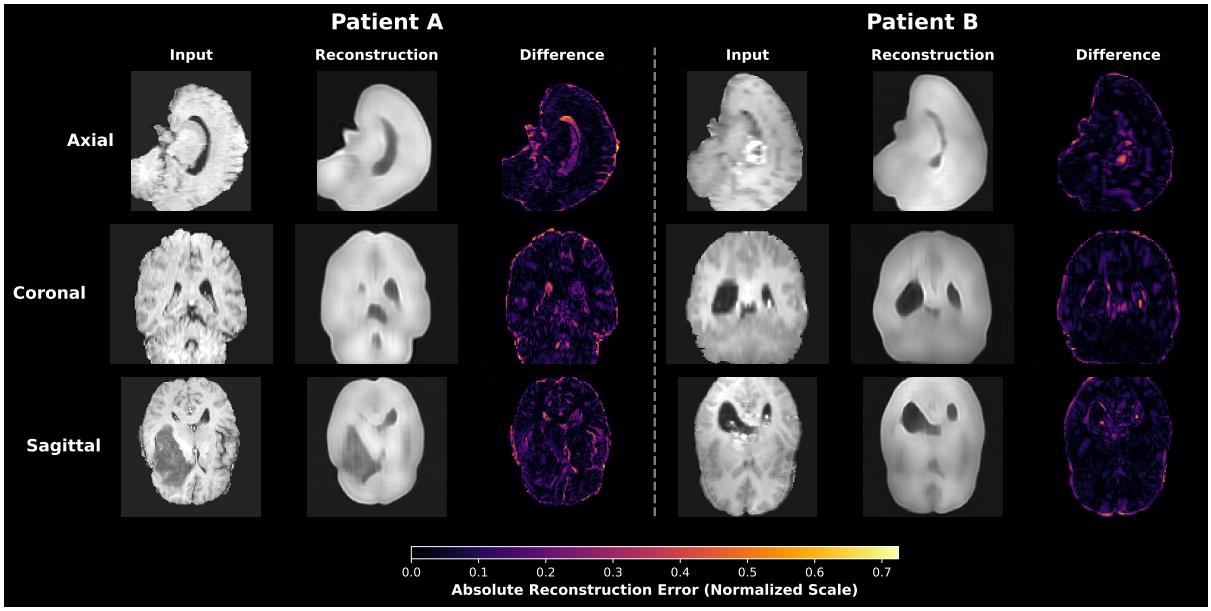


Figure 7: AutoEncoderKL reconstruction results on two validation subjects. From left to right: Input T1ce volume, AutoEncoder reconstruction and absolute difference map shown across axial, coronal and sagittal slices. Difference maps (right column) confirm that the model accurately reconstructs low-frequency volumetric shapes (tumor core, ventricles) but struggles with high-frequency details.

4.3.2 Feature Extraction.

For each patient with available survival data, we extract the latent representation by passing the MRI volume through the trained AutoEncoder and taking the mean of the variational posterior, yielding a tensor $\mathbf{E} \in \mathbb{R}^{C\times H\times W\times D}$ with $C = 3$, $H = 14$, $W = 14$ and $D = 12$. Flattening $\mathbf{E}$ results in a 7056-dimensional feature vector per patient.
As a preliminary analysis of the latent space, we apply Principal Component Analysis (PCA). In contrast to the 3D U-Net encoder, where 98% of the variance was captured by a single component, the AutoEncoder requires 160 components to reach the same threshold, suggesting a less collapsed and more expressive representation.
This observation motivates the hypothesis that survival-relevant information is distributed across both channels and spatial dimensions of the latent tensor. To confirm this, we derive several feature sets from $\mathbf{E}$:

1. **Global Average Pooling.** We compute the mean activation for each channel $c \in \{1, \ldots, C\}$ by averaging across all spatial dimensions:

$$f_c = \frac{1}{H \cdot W \cdot D} \sum_{i=1}^{H} \sum_{j=1}^{W} \sum_{k=1}^{D} \mathbf{E}_{c,i,j,k}.$$

   From these averages, we derive two feature sets:

   (a) **Channel-wise Mean (C0-C1-C2):** The three values are treated as independent scalar features (f_0, f_1, f_2).

   (b) **Global Mean Pooling (3D):** The values are concatenated into a single vector $\mathbf{f}_{\text{pool}} = (f_0, f_1, f_2) \in \mathbb{R}^3$ to preserve inter-channel relationships.

2. **Channel Statistics (9D).** For each channel c, we compute the mean (μ_c), standard deviation (σ_c) and maximum (γ_c) over all spatial locations. These are concatenated to

form:

$$\mathbf{f}_{\text{stats}} = \left[\mu_0, \sigma_0, \gamma_0, \mu_1, \sigma_1, \gamma_1, \mu_2, \sigma_2, \gamma_2\right] \in \mathbb{R}^9.$$

3. **Slice-Based Representations.** To capture spatial variation in the embedding tensor $\mathbf{E}$ while avoiding the high dimensionality of full volumetric representations, we summarize activations using slice-wise profiles along each anatomical axis. For each orientation, we average $\mathbf{E}$ over the two orthogonal spatial dimensions, yielding three profile tensors: $\mathbf{M}_H \in \mathbb{R}^{H\times C}$, $\mathbf{M}_W \in \mathbb{R}^{W\times C}$ and $\mathbf{M}_D \in \mathbb{R}^{D\times C}$. For instance, the profile for the height dimension is defined as:

$$M_H(i,c) = \frac{1}{W \cdot D} \sum_{j=1}^{W} \sum_{k=1}^{D} \mathbf{E}_{c,i,j,k}.$$

   Each matrix is flattened into a vector (e.g., $\mathbf{v}_H \in \mathbb{R}^{HC}$) and, during training, reduced using PCA with 90% variance retention. This yields three components for each of the slice-based representation.

4. **Flattened PCA Representation (10D).** The embedding tensor is vectorized as $\mathbf{e}_{\text{flat}} = \text{vec}(\mathbf{E}) \in \mathbb{R}^{7056}$, standardized and reduced using PCA to a 10-dimensional feature vector.

4.3.3 Classification Pipeline.

The 235 patients are randomly split into training (80%, 188 patients) and test (20%, 47 patients) sets using stratified sampling to preserve class balance in both subsets. To avoid optimistic bias from hyperparameter tuning, we adopt 5-fold cross-validation (CV) on the training set for model selection and hyperparameter optimization. The final selected model is then evaluated once on the held-out test set.
We evaluate five classical models: Logistic Regression, Linear SVM, RBF SVM, Random Forest and Gradient Boosting. Hyperparameter grids are designed to favor strong regularization and shallow architectures, appropriate for small datasets. All features are standardized using `StandardScaler` prior to training. The primary evaluation metric is macro-averaged F1 score, which equally weights both survival classes.

As a critical baseline, we also train all models on age alone (1-dimensional input) and compare performance against the AutoEncoder-derived features. Ablation studies are conducted to assess the individual contribution of each feature set.
Hyperparameter grids used in 5-fold cross-validation for each model are listed in Appendix A.

4.4 Results

Interestingly, the Coronal Slice PCA (3D) representation without incorporating age achieved the highest overall performance, attaining a macro F1 score of 0.656 with a Linear SVM. This outperformed the clinical baseline based solely on age, which reached an F1 score and accuracy of 0.596 using a Random Forest classifier. Although some AutoEncoder-derived features yielded strong results, several configurations, such as individual channel means, underperformed relative to age alone, highlighting the limited prognostic value of global aggregate features. Moreover, combining age with AutoEncoder-derived features did not consistently enhance performance; for example, adding age to the top-performing Coronal PCA representation actually decreased the F1 score to 0.573. Table 7 summarizes the test-set performance (on the held-out 47 patients) across all feature-model combinations, with hyperparameters optimized via cross-validation.

Table 7: Survival prediction performance on BraTS2020. We compare feature representations derived from the $(3 \times 14 \times 14 \times 12)$ embedding tensor $\mathbf{E}$ both as standalone predictors and combined with patient age.

Feature Set	**Age**	**Model**	**Acc.**	**Macro F_1**	**Prec.**	**Rec.**	**F_1 C0**	**F_1 C1**
Clinical Baseline								
Age Only	Yes	Random Forest	**0.596**	**0.596**	0.596	0.596	0.596	0.596
Slice-Based (3D)								
Coronal	No	Linear SVM	**0.660**	**0.656**	0.671	0.662	0.619	0.692
	Yes	RBF SVM	0.574	0.573	0.578	0.576	0.545	0.600
Sagittal	No	Random Forest	0.532	0.530	0.534	0.534	0.500	0.560
	Yes	Linear SVM	0.596	0.595	0.598	0.597	0.578	0.612
Axial	No	Gradient Boosting	0.532	0.532	0.533	0.533	0.522	0.542
	Yes	Gradient Boosting	0.617	0.613	0.626	0.620	0.571	0.654
Global Aggregates								
Flattened PCA (10D)	No	Linear SVM	**0.638**	**0.632**	0.653	0.641	0.585	0.679
	Yes	Logistic Regression	0.553	0.553	0.553	0.553	0.553	0.553
Channel Stats (9D)	No	Gradient Boosting	0.574	0.570	0.581	0.577	0.524	0.615
	Yes	Gradient Boosting	0.574	0.570	0.581	0.577	0.524	0.615
Mean Pooling (3D)	No	RBF SVM	0.553	0.550	0.557	0.555	0.512	0.588
	Yes	RBF SVM	0.574	0.574	0.575	0.575	0.565	0.583
Independent Channels								
Mean (f_0)	No	Logistic Regression	0.532	0.532	0.533	0.533	0.522	0.542
	Yes	Linear SVM	0.574	0.574	0.575	0.575	0.565	0.583
Mean (f_1)	No	Random Forest	0.553	0.552	0.553	0.553	0.571	0.533
	Yes	Linear SVM	0.574	0.574	0.575	0.575	0.565	0.583
Mean (f_2)	No	Logistic Regression	0.532	0.532	0.533	0.533	0.522	0.542
	Yes	Linear SVM	**0.596**	**0.595**	0.598	0.597	0.578	0.612

4.5 Discussion

4.5.1 Representation Quality

The AutoEncoder approach successfully addresses the representation collapse observed in the segmentation bottleneck. While the 3D U-Net required only a single principal component to explain the majority of variance, the AutoEncoder latent space requires 160 components to reach 98% variance explained. This indicates that unsupervised reconstruction learning produces a more distributed representation that captures a broader range of structural information across the latent dimensions.

Several imaging-based feature sets achieved comparable or superior performance to the age-only baseline. The Coronal Slice PCA representation achieved the highest test-set macro F1 score of 0.656, compared to 0.596 for the age-only Random Forest model. This result suggests that the learned latent representations capture tumor characteristics relevant to survival that are not fully explained by age alone.

However, a counterintuitive pattern emerges when combining age with imaging features: in several cases, this combination degrades performance relative to using either modality independently. Since age is a strong standalone predictor and imaging features also show predictive capacity on their own, this degradation cannot be attributed solely to lack of information content in either modality.

A plausible explanation lies in the difficulty of combining different feature types. Age is a scalar, while imaging features are high-dimensional and potentially correlated. With only 188 training samples and strong regularization, the models may fail to find the optimal weighting between these two diverse sources, especially when each provides independent but modest predictive value. The resulting parameter estimates may underutilize both modalities, reducing generalization performance. This suggests a limitation in the modeling approach rather than in the features themselves.

4.5.2 Reconstruction Quality

The quality of learned representations is constrained by reconstruction fidelity. Visual inspection reveals that AutoEncoder reconstructions are noticeably blurry. This is expected given that training was performed from scratch under memory and computational constraints. As a consequence, the learned features likely emphasize coarse, low-frequency structural patterns rather than fine-grained morphological details.

While these global patterns appear sufficient for achieving competitive performance with certain feature configurations, higher reconstruction fidelity could enable the model to capture subtle characteristics that may be more strongly associated with survival outcomes.

4.5.3 Limitations

Single-Modality Training. The AutoEncoder was trained exclusively on T1ce images due to memory constraints. Standard clinical protocols utilize T1, T1ce, T2 and FLAIR in combination, as each modality provides complementary tissue contrast information. Multi-modal training could improve both reconstruction quality and the discriminative capacity of learned features.

Sample Size. The dataset includes only 235 patients with survival information, split into 188 training and 47 test samples. This limited sample size restricts the complexity of models that can be effectively trained and contributes to the observed instability when combining multiple feature sources. With only 188 training samples, regularized classical machine learning models are more appropriate than deep learning approaches, but these simpler models may struggle to find optimal weighting when integrating heterogeneous feature types (clinical scalars

versus high-dimensional imaging features).
The small test set of 47 samples introduces considerable uncertainty in performance estimates. With approximately 23-24 samples per class, differences of even 2-3 misclassifications can substantially shift metrics like F1 score, making it difficult to confidently assess whether observed performance differences reflect genuine model superiority or random variation. Similarly, the 5-fold cross-validation setup partitions the 188 training samples into folds of approximately 38 samples each, which may lead to high variance in fold-level performance and make hyperparameter selection less reliable. Ideally, a larger dataset would enable more stable train-test splits and cross-validation procedures, providing greater confidence in the generalization of results.

4.5.4 Future Directions

This work establishes that unsupervised learning on MRI volumes can produce representations with predictive value for survival outcomes without requiring segmentation annotations. However, several directions could substantially improve performance: s. Finally, more sophisticated feature extraction and fusion strategies could better exploit the spatial structure of the latent representations. Second, exploring alternative architectures or pretraining strategies could improve reconstruction fidelity and enable the capture of finer structural details. Finally, more sophisticated feature extraction and fusion strategies could better exploit the spatial structure of the latent representations.
Despite modest absolute performance, the results validate the feasibility of learning task-relevant representations through unsupervised reconstruction, providing a foundation for future works with larger datasets and resources.

5 Conclusion

This project addressed the clinically relevant tasks of brain tumor segmentation and survival prediction using the BraTS2020 dataset. By enhancing a 3D U-Net with residual connections and Squeeze-and-Excitation modules, we achieved a peak composite Dice score of 81.48%. A significant finding was the "representation collapse" within the segmentation bottleneck, which rendered it unsuitable for survival analysis.
To overcome this, we employed an AutoEncoder-based strategy to learn richer, more distributed latent representations. This approach successfully captured prognostic features that, in several configurations like the Coronal Slice PCA, outperformed the clinical baseline of age alone.
While computational constraints required trade-offs, such as center-cropping volumes and training on single-modality inputs, the results validate the feasibility of unsupervised feature learning for medical prognosis. Future work should prioritize full-resolution multi-modal training and systematic ablation studies to further isolate the impact of specific architectural design choices. Overall, this project demonstrates the potential of combining deep learning with classical machine learning to extract clinically relevant insights from complex medical imaging data.

A Hyperparameter Grids

For reproducibility, we list the hyperparameter values explored during 5-fold cross-validation for each model. The grids were chosen to favor strong regularization and shallow architectures suitable for our relatively small dataset.

Table 8: Hyperparameters explored for each classifier.

Model	**Hyperparameters**
Logistic Regression	Regularization (C): [0.01, 0.1, 1, 10]; Penalty: L2
Linear SVM	Regularization (C): [0.01, 0.1, 1, 10]
RBF SVM	Regularization (C): [0.01, 0.1, 1, 10]; Gamma: [scale, 0.001, 0.01]
Random Forest	Number of trees: [100, 200, 500]; Maximum depth: [None, 5, 10]; Minimum leaf size: [1, 3, 5]
Gradient Boosting	Number of trees: [100, 200]; Learning rate: [0.05, 0.1]; Maximum depth: [1, 2]

References

[1] S. B. et al., "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge," 2019.

[2] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," 2016.

[3] Y. Wu and K. He, "Group normalization," 2018.

[4] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," 2016.

[5] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," 2017.

[6] F. Isensee, P. F. Jaeger, P. M. Full, P. Vollmuth, and K. H. Maier-Hein, "nnu-net for brain tumor segmentation," 2020.

[7] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.

[8] G. Lozupone, A. Bria, F. Fontanella, F. J. A. Meijer, C. D. Stefano, and H. Huisman, "Latent diffusion autoencoders: Toward efficient and meaningful unsupervised representation learning in medical imaging," 2025.

[9] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," 2018.

[10] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," 2016.

[11] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," 2018.

[12] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡0.5mb model size," 2016.

[13] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," 2017.

[14] W. H. Pinaya, P.-D. Tudosiu, J. Dafflon, P. F. Da Costa, V. Fernandez, P. Nachev, S. Ourselin, and M. J. Cardoso, "Brain imaging generation with latent diffusion models," in *MICCAI Workshop on Deep Generative Models*, pp. 117–126, Springer, 2022.