

ROI-Aware Deep Learning Framework For Brain MRI Compression

**Thesis Submitted
in Partial Fulfillment of the Requirements for the
Degree of**

**MASTER OF TECHNOLOGY
in
Signal Processing and Digital Design**

**by
Sidharth Sehrawat
(Roll No. 2K24/SPD/19)**

**Under the Supervision of
Dr. N. Jayanthi**
Associate Professor,

Department of Electronics and Communication Engineering



**Department of Electronics and Communication Engineering
DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)
Shahbad Daultpur, Main Bawana Road, Delhi-110042, India

May, 2026



DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daultpur, Main Bawana Road, Delhi-42

ACKNOWLEDGEMENT

I would like to begin by expressing my sincere gratitude to my supervisor, Dr. N. Jayanthi, for her continuous guidance, feedback and patience through the course of this thesis. She showed me the path to achieve the targets by helping me break the work into manageable pieces, explained the bigger picture whenever I lost track, and gave me the freedom to try out ideas while still steering me when things went off the rails. Without her constant support and motivation, this thesis would not have come together.

Sidharth Sehwat
(2K24/SPD/19)



DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I **Sidharth Sehrawat** hereby certify that the work which is being presented in the thesis entitled **ROI-Aware Deep Learning Framework for Brain MRI Compression** in partial fulfillment of the requirements for the award of the Degree of Master of Technology, submitted in the Department of **Electronics and Communication Engineering**, Delhi Technological University is an authentic record of my own work carried out during the period from **August 2024** to **May 2026** under the supervision of **Dr. N. Jayanthi**, Associate Professor, Department of Electronics and Communication Engineering, Delhi Technological University.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

Signature of External Examiner

Signature of Supervisor



DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR

Certified that **Sidharth Sehrawat (2K24/SPD/19)** has carried out the research work presented in this thesis entitled **ROI-Aware Deep Learning Framework for Brain MRI Compression** for the award of **Master of Technology** from the Department of Electronics and Communication Engineering, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work and studies are carried out by the student herself/himself, and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Signature

Dr. N. Jayanthi
Associate Professor
(Department of Electronics & Communication Engineering, DTU)

Date: _____

ABSTRACT

Magnetic Resonance Imaging (MRI) has become a routine tool for diagnosing brain conditions, especially tumours. One thing that comes with MRI's clinical usefulness is a real practical problem: the scans take up a lot of storage. A single patient study can run into hundreds of megabytes, and hospitals deal with thousands of such studies every year. What makes medical compression different from regular image compression is that we cannot afford to blur out the very region a radiologist is going to look at. So we need a compression scheme that knows what to protect and what to compress.

In this thesis, we begin by surveying the existing landscape of image compression, both classical (JPEG, JPEG2000) and learning-based (autoencoder-style codecs), and the work that has been done on Region-of-Interest (ROI) compression for medical images. Following this, we present a deep-learning-based ROI-aware compression pipeline for brain MRI. The pipeline has three connected stages: a U-Net segmentation model identifies the tumour region from each axial FLAIR slice, the predicted mask is expanded by a controllable safety margin to absorb small segmentation errors, and a convolutional autoencoder compresses everything outside the safe region while the ROI itself is kept intact. The final image is reconstructed by stitching the original ROI back with the decoded background.

Next, we add two ideas that make the framework more practical. We define an adaptive compression-ratio rule that scales the background CR (16:1, 32:1 or 64:1) based on the relative ROI size in each slice. And we run a margin sweep over $m = 0, 2, 4, 6, 8$ and 12 pixels to map how the trade-off between compression efficiency and diagnostic safety actually behaves as the ROI is expanded. The second of these is also the central contribution of the conference paper communicated from this work, for which we revised the autoencoder design and trained it with an ROI-aware loss.

Finally, we report what the framework actually achieves on the BraTS 2020 dataset. The U-Net reaches a mean Dice of 0.7795 and recall of 0.8440. The full system achieves a PSNR of 29.78 dB, SSIM of 0.8753, a mean effective compression ratio of 25.50:1 with a maximum of 64:1, all while keeping the tumour region bit-exact by construction. The margin sweep shows that ROI coverage improves from 0.756 ($m = 0$) to 0.982 ($m = 12$), with leakage dropping from 0.244 to 0.018 at a controlled cost to global PSNR. The practical operating range works out to $m = 6$ to 8 pixels. We close the thesis with a discussion of limitations, concrete future directions, and the social implications of deploying this kind of system in real clinical settings.

LIST OF PUBLICATIONS

The following research papers have been communicated from the work presented in this thesis:

- [1] Sidharth Sehrawat and N. Jayanthi, “*ROI-Aware Deep Learning Framework for Brain MRI Compression*,” communicated to a peer-reviewed conference, 2026.
- [2] Sidharth Sehrawat and N. Jayanthi, “*Margin-Aware ROI Expansion for Robust Brain MRI Compression under Segmentation Uncertainty*,” communicated to a peer-reviewed conference, 2026.

TABLE OF CONTENTS

Title.....	i
Acknowledgement.....	ii
Candidate's Declaration.....	iii
Certificate by the Supervisor.....	iv
Abstract	v
List of Publications	vi
Table of Contents	vii
List of Tables.....	x
List of Figures	xi
List of Symbols, Abbreviations and Nomenclature	xii
CHAPTER 1: INTRODUCTION.....	1
1.1 Brain MRI and the Need for Compression.....	1
1.2 Region-of-Interest Compression	2
1.3 Segmentation and U-Net	2
1.4 Why a Safety Margin	3
1.5 Motivation	3
1.6 Problem Statement	4
1.7 Objectives of the Thesis	4
1.8 Contributions	4
1.9 Organisation of the Thesis.....	4
CHAPTER 2: LITERATURE REVIEW.....	6
2.1 Classical Image Compression	6
2.1.1 JPEG and the DCT.....	6
2.1.2 JPEG2000 and the wavelet transform.....	6
2.1.3 Other classical schemes	6
2.2 Compression Issues Specific to Medical Imaging	7
2.2.1 Regulatory and Diagnostic Constraints.. ..	7
2.2.2 Bit Depth.....	7
2.2.3 Three-dimensional structure	7
2.2.4 Recent Surveys	7
2.3 Region-of-Interest Based Compression.....	8
2.3.1 Manual Roi Selection.....	8
2.3.2 Automatic ROI selection	8
2.3.3 Hybrid Lossless/Lossy Schemes.....	8

2.4 Learning-Based Image Compression	9
2.4.1 Balle and Variational codec family	9
2.4.2 Attention and Hybrid Architectures.....	9
2.4.3 Learned Compression for Medical Images	9
2.5 U-Net and Brain Tumour Segmentation	10
2.5.1 U-Net Architecture	10
2.5.2 BraTS and Brain Tumour segmentation	10
2.5.3 Segmentation Uncertainty	10
2.6 Identified Research Gaps.....	11
CHAPTER 3: DESIGN AND METHODOLOGY	12
3.1 Overview of the Proposed Framework.....	12
3.2 Dataset	13
3.3 Data Preprocessing	14
3.4 Tumour Segmentation Using U-Net.....	15
3.4.1 Network architecture.....	15
3.4.2 Dice-focal loss function.....	16
3.5 Safety Margin via Morphological Dilation	17
3.6 Background Compression	17
3.6.1 Initial autoencoder design: stride-2 bottleneck.....	18
3.6.2 Reconstruction loss for the initial design.....	19
3.6.3 Refined autoencoder design.....	19
3.6.4 ROI-aware reconstruction loss for the refined design	20
3.7 Adaptive Compression Ratio.....	21
3.8 Image Reconstruction.....	21
3.9 Evaluation Metrics	22
CHAPTER 4: RESULTS AND DISCUSSION	24
4.1 Hardware and Software Environment	24
4.2 Data Loading and Training Trajectories	25
4.2.1 Data loading.....	25
4.2.2 U-Net training	25
4.2.3 Autoencoder training	26
4.3 Running the Models	26
4.4 Segmentation and Reconstruction Results	28
4.5 Adaptive Compression Ratio Distribution	28
4.6 Margin Sweep Results.....	29
4.7 Coverage-Leakage-Quality Trade-off.....	31

4.8 Comparison with Prior Work	32
4.9 Qualitative Observations	33
4.10 Discussion	33
4.11 Limitations.....	34
CHAPTER 5: CONCLUSION AND FUTURE SCOPE	35
5.1 Conclusion.....	35
5.2 Future Scope.....	36
5.3 Social Impact.....	37
REFERENCES.....	38
APPENDIX-I: SELECTED SOURCE CODE LISTINGS	41
APPENDIX-II: SUPPLEMENTARY RESULTS AND FIGURES.....	44
LIST OF PUBLICATIONS AND THEIR PROOFS.....	46
PLAGIARISM REPORT	47
CURRICULUM VITAE / BRIEF PROFILE.....	48

LIST OF TABLES

Table 3.1	Preprocessing parameters used on BraTS 2020 slices.....	14
Table 3.2	Layer-wise specification of the U-Net used for tumour segmentation. ..	15
Table 3.3	Initial autoencoder: stride-2 convolutional design with $16 \times 16 \times 4$ latent.	18
Table 3.4	Refined autoencoder: MaxPool / UpSample design used for the margin sweep study.	20
Table 3.5	Adaptive compression ratio assigned per slice based on relative ROI area.	21
Table 4.1	Software stack used in this work.....	24
Table 4.2	Hyper-parameters for the U-Net and both autoencoder variants.	27
Table 4.3	Segmentation, reconstruction and adaptive compression results.....	28
Table 4.4	Margin sweep results across $m \in \{0, 2, 4, 6, 8, 12\}$	30
Table 4.5	Indicative comparison with classical and learned codecs.	32
Table II.1	Per-patient coverage and leakage statistics at margin $m = 6$	44

LIST OF FIGURES

Fig. 3.1	Block diagram of the proposed ROI-aware compression pipeline.....	12
Fig. 3.2	Sample axial FLAIR slice with its ground-truth tumour mask overlaid....	14
Fig. 3.3	U-Net architecture used in this work, with two pooling levels with skip connections.....	16
Fig. 3.4	Effect of morphological dilation on a predicted tumour mask.	17
Fig. 3.5(a)	Initial autoencoder, stride-2 bottleneck design.....	19
Fig. 3.5(b)	Refined autoencoder, MaxPool / UpSample design with ROI-aware MSE loss.	20
Fig. 4.1	Distribution of slices across the three CR buckets.	29
Fig. 4.2	Coverage and leakage versus dilation margin.	30
Fig. 4.3	Global PSNR and SSIM versus dilation margin.....	31
Fig. 4.4	Combined trade-off score across margins, with practical operating range highlighted.....	32
Fig. II.1	Reconstruction examples at four margin values for four test slices.....	45

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

Abbreviations

AE	Autoencoder
BraTS	Brain Tumour Segmentation challenge
CNN	Convolutional Neural Network
CR	Compression Ratio
CT	Computed Tomography
DCT	Discrete Cosine Transform
DICOM	Digital Imaging and Communications in Medicine
DWT	Discrete Wavelet Transform
FLAIR	Fluid-Attenuated Inversion Recovery
GPU	Graphics Processing Unit
JPEG	Joint Photographic Experts Group
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
PACS	Picture Archiving and Communication System
PSNR	Peak Signal-to-Noise Ratio
ROI	Region of Interest
SSIM	Structural Similarity Index Measure
T1, T2	T1-weighted, T2-weighted MRI modalities
U-Net	U-shaped convolutional segmentation network

Symbols

X	Input MRI slice
M	Binary tumour segmentation mask
$\hat{M}$	Predicted tumour mask from U-Net
M_{safe}	Mask after safety-margin dilation
m	Dilation margin in pixels
X_{bg}	Background image: $(1 - M_{safe}) \odot X$
z	Latent representation from the autoencoder encoder
E, D	Encoder and decoder networks
$\hat{X}$	Reconstructed image
r	Relative ROI area
τ_{low}, τ_{high}	Thresholds for the adaptive CR rule (0.20 and 0.40)
$CR_{low}, CR_{mid}, CR_{high}$	Background compression ratios (16, 32, 64)

$\oplus$	Morphological dilation operator
$\odot$	Element-wise (Hadamard) product
L_{Dice}, L_{AE}	Dice loss and autoencoder reconstruction loss
λ_{bg}	Background-term weight in the ROI-aware loss (0.2)

CHAPTER 1

INTRODUCTION

Medical Imaging is important in today's healthcare. Any time a structural problem is suspected inside the body, some kind of imaging study is suggested. For brain conditions specifically, Magnetic Resonance Imaging (MRI) is generally first choice. It does not use ionising radiation. Gives strong soft-tissue contrast, and offers multiple pulse sequences (T1, T1-contrast, T2, FLAIR) that bring out different anatomical features. For brain tumours, which are the focus of this thesis, FLAIR is the most informative single modality, which suppresses fluid signal and makes lesions stand out clearly [8].

It is seen that a single brain study can generate four 3D volumes, each with well over 150 slices at 240×240 or higher resolution. That becomes a problem as there are a lot of patients and individuals, so storage cost increases at a huge scale. Bandwidth follows the same curve. Moving scans between sites for second opinions, or pushing them through a Picture Archiving and Communication System (PACS), takes quite a lot of time. Compressing these images is therefore practically necessary, but doing it well requires more thought than a regular photograph would need [9][27].

1.1 Brain MRI and the Need for Compression

MRI images are different from natural photographs:

- They have diagnostic information. A small detail near a tumour can change a treatment plan.
- They have higher bit-depth (12 or 16 bits per pixel) than standard image formats expect.
- They are usually 3D volumes, not flat 2D pictures, so there is more correlation between slices than between two random images.
- They get archived for long periods. Patient records often need to be kept for years.

Plain lossless compression preserves every pixel but is bounded; ratios above 4:1 are rare. Lossy compression like JPEG2000 reaches much higher ratios and is part of the DICOM standard, but applied blindly it can blur exactly the regions a radiologist would look at. So we need something in between, a method that knows what to protect and what to compress [9] [22] [26].

1.2 Region-of-Interest Compression

The idea behind Region-of-Interest (ROI) based compression is simple. We split image into two pieces. The first piece is the part that's important for diagnosis. In our case the tumour and a small buffer of surrounding tissue. The second is the rest, which is mostly healthy parenchyma, ventricles, skull and background. The first piece is kept at high fidelity, while second one is compressed aggressively. The overall compression ratio depends on how much of the slice falls into each bucket [29].

The benefits:

- Diagnostic content is preserved by construction, not by hope.
- Stronger compression can be applied to background, lifting the overall ratio without touching the diagnostic region.
- The system gives a clinician an audit trail: the ROI mask, the margin used, and the resulting safe region can all be inspected after the fact.

The hard part is figuring out where the ROI actually is. Earlier ROI methods relied on radiologists drawing the boundary by hand. That works but does not scale, and two experts will draw slightly different boundaries on the same slice. A more practical approach is to detect the region automatically from the image. This is where deep learning steps in [6] [24] [28].

1.3 Segmentation and U-Net

A segmentation model takes an image and assigns a label to every pixel. In our case, “tumour” or “not tumour”. The output is a mask the same shape as the input image. Modern segmentation models are convolutional neural networks (CNNs), and for biomedical images the network of choice is U-Net.

U-Net was introduced by Ronneberger, Fischer and Brox in 2015 [20]. Its main features are:

- “U”-shaped encoder-decoder structure. The encoder downsamples the input and increases feature depth, the decoder upsamples back to the original resolution.
- Skip connections between matching encoder and decoder levels, so fine spatial details are not lost during downsampling.
- Trained end-to-end on segmentation masks, with losses like Dice or cross-entropy.

In this thesis we use a small U-Net with two downsampling levels and filter counts $32 \rightarrow 64 \rightarrow 128$, suited for 128×128 FLAIR slices. The network is deliberately kept

light because it is one block in a longer compression pipeline and has to run fast in practice.

1.4 Why a Safety Margin

Deep segmentation models, even good ones, are not perfect. They tend to over-segment in some slices and under-segment in others. The dangerous case for a compression pipeline is under-segmentation. A few real-tumour pixels get marked as background, then get crushed by the codec, and that information is gone. A purely segmentation-driven compressor inherits every mistake the segmenter makes [14] [15] [32].

To deal with this, we expand the predicted mask before passing it to the codec. The expansion is done with a morphological dilation using a structuring element of radius m pixels. The choice of m controls how much safety we get:

- Small m : less margin, higher overall compression, more risk of missing real tumour pixels.
- Large m : more margin, lower overall compression, less risk of missing tumour pixels.

We do not try to pick the “right” m theoretically. Instead we run a margin sweep over $m \in \{0, 2, 4, 6, 8, 12\}$ and report what happens at each value. The reader (or the deploying institution) can then read off the operating point that matches their tolerance for residual leakage.

1.5 Motivation

Hospital storage budgets do not grow as fast as imaging data. Public hospitals in lower-resource settings, where one MRI machine still produces a steady stream of scans, feel this even more sharply. A compression scheme that gives a high ratio without disturbing the diagnostic region has immediate and practical value. Traditional codecs like JPEG and JPEG2000 are content-agnostic [21] [26]. They treat the whole image the same way. That is wasteful for a brain MRI, because most of the slice is healthy tissue or background that only needs enough fidelity for orientation, while a smaller region needs nearly lossless reconstruction. Content-aware methods promise a better trade-off, but the question of how to define content automatically is exactly what deep learning lets us answer. Most published ROI compression methods either rely on a human to mark the region, or assume that the automatic segmentation is essentially correct [6] [14] [28]. Neither of these holds in practice. So we wanted to design a

pipeline that explicitly accounts for an imperfect segmenter and gives the user a tunable knob to trade safety against efficiency. That knob is the margin.

1.6 Problem Statement

Medical images are big in size. Single MRI study can be of hundreds of megabytes, and hospitals deal with a lot of them. Standard compression might be not suitable in this case. But existing ROI-based methods mostly need someone to manually mark the region. This removes the automation. The learning-based codecs that do exist treat every pixel the same. And even the ones that do try ROI compression don't really handle the fact that segmentation models make mistakes. Small masking slightly off position can result in the tumor region being altered. This can result in improper diagnosis. So the problem becomes clear that we need a pipeline that finds the tumour itself, protects it with some room for error, and still compresses the background hard enough to actually be useful. We need to find a solution that can be adopted without much need of compute force and complexity [3] [4] [6] [9] [14].

1.7 Objectives of the Thesis

- To develop an ROI-aware deep learning framework for brain MRI compression by combining U-Net based tumour segmentation with adaptive background compression.
- To improve diagnostic safety during compression using safety-margin based ROI protection and evaluate the trade-off between compression efficiency and reconstruction quality.

1.8 Contributions

We were able to achieve the following contributions for our respective cases:

- Full ROI-aware compression pipeline for brain MRI that combines automatic segmentation, a tunable safety margin and learned background compression in one end-to-end flow.
- A revised autoencoder design for the margin-aware study, with an ROI-aware MSE loss that handles the masked-input scenario more gracefully than plain MSE.
- A systematic margin sweep mapping the trade-off between compression efficiency and ROI safety.

1.9 Organisation of the Thesis

The rest of the thesis is as follows:

- **Chapter 2, Literature Review.** Reviews classical and learning-based image compression, then the ROI-based methods used in medical imaging, the U-Net family of segmentation networks, and the work on segmentation uncertainty. It ends by setting out the gaps this thesis tries to address.
- **Chapter 3, Design and Methodology.** Goes through the pipeline one stage at a time: the dataset, preprocessing, the U-Net architecture and its loss, the safety-margin step, both autoencoder versions (the first design and the refined one), the adaptive ratio rule, the fusion step, and the evaluation metrics. The equations and architectural details are kept in this chapter.
- **Chapter 4, Results and Discussion.** Reports the experimental results. The first half covers segmentation accuracy and the main reconstruction numbers under the adaptive compression-ratio policy. The second half is the margin sweep study, which shows how coverage, leakage and reconstruction quality change as the safety margin is varied. The chapter ends with a comparison against earlier work, a few qualitative observations, and a discussion of why the framework behaves the way it does.
- **Chapter 5, Conclusion and Future Scope.** Summarises what was built, states the limitations, and lists concrete future directions, from multi-modal and 3D extensions to uncertainty-driven spatial margins and clinical validation.

References, the source-code appendix, supplementary results, the list of publications, the plagiarism report and the candidate's brief profile follow at the back of the document.

CHAPTER 2

LITERATURE REVIEW

In this chapter we study the prior work that helped us in this thesis. We start with classical image compression, then try to focus on the medical-imaging context, then move to ROI-based methods, then to learning-based compression, and finally to U-Net and segmentation uncertainty. The chapter closes with the gaps we found in the literature and the problem we set up to address them.

2.1 Classical Image Compression

Image compression is essential in digital imaging if we want to reduce storage and transmission costs.. The two basic ideas are to transform coding (mapping pixels to a basis where energy concentrates in a few coefficients) and entropy coding (using fewer bits for more common symbols) [21].

2.1.1 JPEG and the DCT

JPEG uses the Discrete Cosine Transform (DCT) on 8×8 blocks. Each block is transformed into the frequency domain, the coefficients are quantised, and the result is packed with Huffman or arithmetic coding [1] [21]. JPEG was designed for natural photographs and works well there. The main weaknesses for medical imaging:

- Baseline JPEG assumes 8-bit data, but most MRI scanners produce 12- or 16-bit images.
- Quantisation across the whole image means small but diagnostically important features can blur out alongside cosmetic details.
- Block boundaries can become visible at high compression, a problem when the boundary cuts across a lesion [9].

2.1.2 JPEG2000 and the Wavelet Transform

JPEG2000 replaced block-DCT with the Discrete Wavelet Transform (DWT). DWT gives a multi-resolution view of the image and produces a quality curve that scales smoothly with bit-rate. JPEG2000 supports higher bit-depths natively and is part of the DICOM standard. It also supports an explicit ROI mechanism where one or more regions can be coded at higher fidelity, but the user has to mark the region in advance, which is a manual step [26].

2.1.3 Other Classical Schemes

Other than JPEG and JPEG2000 there are several variants :

- Arithmetic-coded JPEG: better entropy coding, marginal compression gain.
- JPEG-LS: a near-lossless variant with a bounded per-pixel error.
- SPIHT and EZW: wavelet-domain progressive coding methods that pre-date JPEG2000.

A more broad survey of these methods is provided by Ungureanu et al. [29] and by Al-jawaherry and Hamid [1]. The classical codecs are mature, but none of them say region is diagnostically important. They were not designed with medical images in mind.

2.2 Compression Issues Specific to Medical Imaging

2.2.1 Regulatory and Diagnostic Constraints

Lossy compression is allowed in DICOM only when its use is justified and documented. There are reported cases where subtle pathology was missed after aggressive compression [9]. A loss that is invisible at global PSNR can still be very visible inside a tumour boundary.

2.2.2 Bit Depth

Many CT and MRI scans are 12 or 16 bits per pixel. Schemes that assume 8-bit data either downcast (losing information) or extend awkwardly (losing compression efficiency). JPEG2000 handles this natively, which is one reason it dominates the DICOM landscape [9] [26].

2.2.3 Three-dimensional Structure

An MRI is a stack of correlated slices, not an isolated image. Any 2D scheme leaves inter-slice correlation on the table. Volumetric codecs and 3D wavelets exist, but they are heavier and harder to deploy. In this thesis we deliberately use a 2D pipeline for tractability, while flagging the limitation [10] [16].

2.2.4 Recent Surveys

A broad treatment of the challenges in medical-image processing is given by Deheyab et al. [8]. A more compression-focused survey covering both traditional and learning-based methods is provided by Tong et al. [27]. Vijaya et al. [30] study a hybrid lossless/lossy approach for medical images. Gopal et al. [10] catalogue further variants. Jamil [13] reviews the quality-assessment metrics used specifically for compressed images, which is relevant when comparing different codecs.

2.3 Region-of-Interest Based Compression

ROI-based methods sit between strictly lossless and aggressively lossy compression. The image is split into an ROI and a background; the ROI gets high-fidelity coding, the background gets lossy coding. The overall ratio depends on the relative size of the ROI [24] [29].

2.3.1 Manual ROI Selection

Older work needed a radiologist or technician to draw the region. The encoder used this mask to bias its bit allocation. It is conceptually clean but:

- Slow. Every scan needs a human in the loop.
- Inconsistent. Two experts will draw slightly different boundaries on the same slice.
- Hard to audit. The same scan can end up compressed differently depending on who handled it.

Bharath et al. [6] discuss lossless ROI compression for MRI with added security guarantees, and report that automating the region detection is a recurring practical issue.

2.3.2 Automatic ROI Selection

Several authors moved towards semi-automatic and fully automatic ROI extraction. Sran et al. [24] use visual saliency to detect important regions in brain MRI and apply segmentation-based compression. Tunga and Singh [28] go further and automatically extract the tumour region from the MRI itself before compressing everything else.

2.3.3 Hybrid Lossless/Lossy Schemes

Hybrid methods apply trully lossless coding inside the ROI and stronger lossy coding outside. Advantage is that no information is changed in diagnostic region while the disadvantage is that the lossless side of the trade is bounded. One can only compress lossless tissue so much before you stop adding value. The trade-off between strict-inside guarantee and overall compression has to be made one way or another [22] [30].

2.4 Learning-Based Image Compression

2.4.1 Ballé and the Variational Codec Family

Ballé et al. [3] showed that an end-to-end optimised autoencoder, with a careful entropy model, could match or beat JPEG2000 on natural images. The architecture is simple:

- An encoder network compresses the image into a low-dimensional latent code.
- A probability model is fitted over the latents so they can be arithmetically coded.
- A decoder reconstructs the image from the (quantised) code.
- Training optimises a rate-distortion loss of the form $D + \lambda R$.

Their follow-up paper [4] introduced a hyperprior, a small extra network that models spatial correlations in the latent representation, and pushed the rate-distortion curve further.

2.4.2 Attention and Hybrid Architectures

Several improvements appeared on top of this template. He et al. [11] add a lightweight attention mechanism to push real-time performance. Dai [7] uses cross-attention combined with the wavelet transform and reports gains over JPEG2000 baselines specifically for medical compression. A critical review of these architectures is given by Mishra et al. [19]. Huang and Wu [12] survey the full state of end-to-end learned image compression and argue that practical deployment is now closer than it used to be.

2.4.3 Learned Compression for Medical Images

Al-Khafaji and Ramaha [2] propose a hybrid deep architecture for scalable image compression with medical applications. Sharma et al. [22] propose a block-adaptive near-lossless scheme that bounds per-pixel error and report good diagnostic-quality results. A method-of-color-images compression study by Spasova and Boychev [23] and a machine-learning oriented overview by BH and Prakash [5] round out the picture.

One pattern is consistent across all of these works: the rate-distortion loss does not, by itself, know which regions are diagnostically important. The loss treats every pixel equally. Some authors weight the loss to push the network towards better quality in important regions, but a more direct route is to combine the learned compressor with an explicit segmentation pass. That is the path this thesis follows.

2.5 U-Net and Brain Tumour Segmentation

2.5.1 The U-Net Architecture

U-Net was introduced by Ronneberger, Fischer and Brox [20] for biomedical image segmentation. The architecture has three main features:

- “U” shape: a contracting path (encoder) followed by an expanding path (decoder).
- Skip connections that copy feature maps from each encoder level to the matching decoder level at the same spatial resolution.
- Symmetric downsampling and upsampling, so the output has the same spatial resolution as the input.

The skip connections are critical. They let the decoder recover fine spatial detail that pooling would otherwise lose. U-Net was originally trained on very small biomedical datasets and still produced strong segmentations, which made it spread quickly across the field.

2.5.2 BraTS and Brain-Tumour Segmentation

For brain tumour segmentation the standard benchmark is BraTS [17] [18]. The challenge provides multi-modal MRI scans and expert-annotated tumour masks and many U-Net variants have been tried on BraTS: 3D U-Net, attention U-Net, nnU-Net, transformer encoders. State-of-the-art Dice on BraTS 2020 sits around 0.85 – 0.91 for whole-tumour segmentation when 3D networks and all four modalities are used. The broader landscape of deep-learning methods in medical image analysis is reviewed by Lee and Fujita [16].

In this thesis we deliberately use a lightweight 2D U-Net on FLAIR only, working on 128×128 slices. We trade a few Dice points for a much smaller and faster network, then absorb the gap with a downstream safety margin. The trade-off is justified in Chapter 3.

2.5.3 Segmentation Uncertainty

Deep segmentation models produce probabilities, not hard labels. A pixel is classified as tumour or background only after thresholding the probability map. Around 0.5 the threshold is balanced. Below 0.5 it is recall-biased. The model marks more pixels as tumour, including some that are not, but misses fewer real-tumour pixels. The opposite holds above 0.5.

Beyond thresholding there is a deeper question: how confident is the model about any given prediction? Kendall and Gal [15] proposed a framework for capturing

aleatoric and epistemic uncertainty in Bayesian deep learning. Jungo and Reyes [14] looked specifically at how reliable uncertainty estimates are for medical segmentation, and showed that the estimates themselves can be poorly calibrated.

The link to compression is direct. If the segmentation is uncertain about a boundary pixel, and that segmentation drives compression, the uncertainty leaks straight into the codec. A pixel marked background gets compressed; if it was actually tumour, that information is lost. This thesis takes a simpler route than full uncertainty modelling. The ROI is expanded by a morphological dilation of fixed radius. The dilation gives a deterministic buffer against the kind of small boundary errors that uncertainty would otherwise have to be modelled to fix.

2.6 Identified Research Gaps

1. Most ROI-based methods either need a human to mark the region (does not scale) or assume the automatic segmenter is essentially perfect (it is not) [6] [26] (JPEG2000 ROI needs manual marking). There is a gap for methods that explicitly handle imperfect segmentation [14] [15]
2. Learning-based codecs optimise a global rate-distortion loss that treats every pixel equally [3][4]. Diagnostic content is not part of the loss. Combining a learned background codec with an explicit segmentation pass gives a cleaner separation. The segmenter decides what is important, the codec decides how to compress what is not. [7] [11].
3. Slice-level adaptive control of the compression ratio is rarely explored [10] [27]). Tumour size varies from slice to slice in the same patient, but most methods apply a single fixed CR. A simple rule based on relative ROI area can lift mean compression without giving up diagnostic safety [22] [24] [28].
4. The trade-off between compression efficiency and ROI safety has not been mapped out systematically. A margin sweep is one of the contributions of this work precisely because, before running it, we did not know where the sweet spot sits [27][29].
5. Evaluation metrics for ROI-aware compression are not yet standard. Coverage and leakage refer to how much of the true tumour stays inside the protected region and how much spills outside, are direct, intuitive numbers that go beyond global PSNR/SSIM [25] [31].

CHAPTER 3

DESIGN AND METHODOLOGY

In this chapter we describe the proposed framework piece by piece. We begin with an overview of the full pipeline, then walk through the dataset, preprocessing, U-Net segmenter, safety margin step, both autoencoder variants (the initial design and the refined design), the adaptive compression-ratio rule, the fusion step and finally the evaluation metrics. All equations and architectural details live in this chapter; the experimental results that come out of running this pipeline are reported in Chapter 4.

3.1 Overview of the Proposed Framework

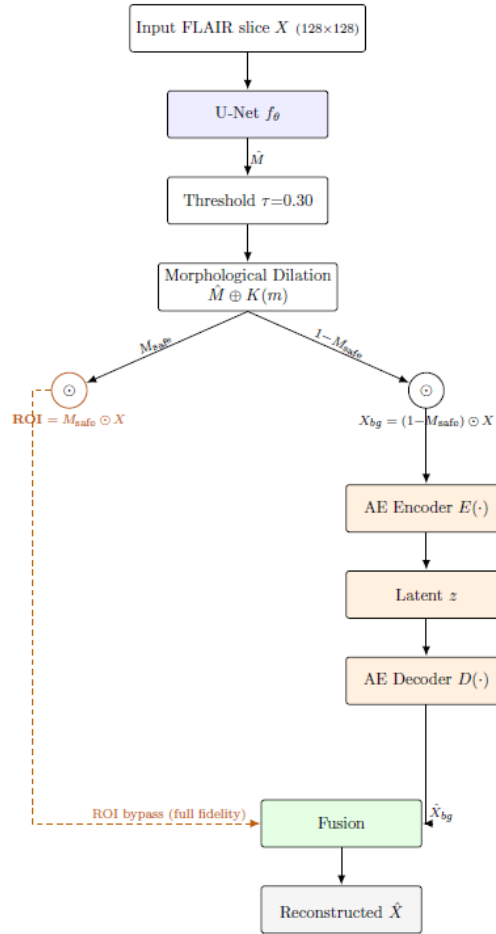


Fig. 3.1 Block diagram of the proposed ROI-aware compression pipeline. The dashed arrow indicates that ROI pixels bypass the autoencoder and are stitched back unchanged at the fusion step.

The pipeline does three things, in order. We find the tumour. We put a buffer around it. We compress what is left. None of these is a new idea on its own, but putting them in one pipeline with a slice-level rate-control rule on top is where this thesis tries to add value.

The input is a single axial FLAIR slice at 128×128 . A U-Net produces a per-pixel tumour probability map. We threshold the map at 0.30 (recall-biased) and then expand the mask with a morphological dilation of radius m . The expanded mask defines the protected region. Everything outside it is fed into a convolutional autoencoder. The autoencoder's output is then stitched back together with the original ROI pixels. A small bookkeeping rule chooses how aggressively to code the background based on how much of the slice was tumour [3] [20] [29].

3.2 Dataset

All experiments use BraTS 2020. It is the most widely used benchmark for brain tumour segmentation and has been kept up to date for over a decade [17]. The 2020 release ships with 369 training subjects, each containing four MRI modalities (T1, T1ce, T2, FLAIR) and a hand-annotated tumour mask. The masks separate necrotic core, enhancing tumour and peritumoural oedema; in this work we merge all three into a single binary whole-tumour label.

We use only the FLAIR modality. The reasons are:

- FLAIR gives strong contrast between tumour and surrounding tissue, which makes whole-tumour segmentation feasible from a single modality.
- Using one modality keeps memory, training time and integration overhead low, which matters when the segmenter is just one block in a longer pipeline.
- The cost is a few Dice points relative to a multi-modal 3D network. The downstream safety margin absorbs most of that loss.

Scan volumes are $240 \times 240 \times 155$. We work slice by slice on the axial direction. A 2D pipeline loses some inter-slice context but lines up with how a radiologist actually scrolls through the volume, one slice at a time, and keeps the parameter count manageable [16].

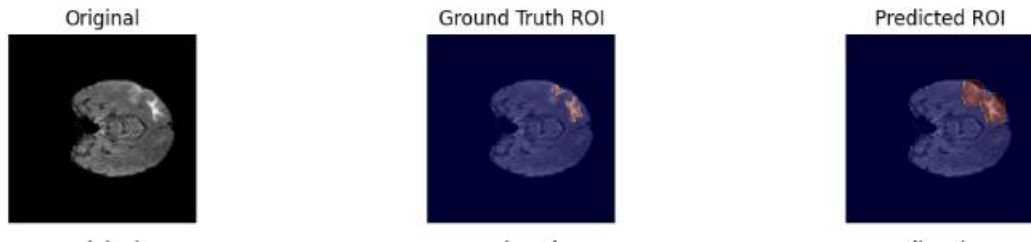


Fig. 3.2 Sample axial FLAIR slice from BraTS 2020 with its ground-truth tumour mask overlaid.

3.3 Data Preprocessing

Preprocessing converts raw NIfTI volumes into a clean tensor dataset. The main steps are:

1. Load each volume with NiBabel as a NumPy array.
2. Walk through axial slices with a configurable step (2 during U-Net training, 3 during evaluation. Every n-th slice is read, to skip near-duplicates).
3. Resize each slice from 240×240 to 128×128 with bilinear interpolation; resize the matching mask with nearest-neighbour interpolation.
4. Normalise intensities to $[0, 1]$ per slice.
5. Binarise the mask: any positive BraTS label is collapsed into a single foreground class.
6. Drop slices with fewer than 10 foreground pixels (these are the first/last few axial slices where the tumour barely intersects the cross-section).

The normalisation and binarisation rules in equation form:

$$I_{norm}(x, y) = [I(x, y) - I_{min}] / [I_{max} - I_{min} + \epsilon] \quad (3.1)$$

$$M(x, y) = 1 \quad \text{if } S(x, y) > 0, \quad \text{else } 0 \quad (3.2)$$

After this filter, roughly half the candidate slices per subject survive, which is enough for the U-Net to converge cleanly [17]. Table 3.1 summarises the parameters.

Table 3.1 Preprocessing parameters used on BraTS 2020 slices.

Parameter	Value	Purpose
Slice resolution	128×128	Keeps memory and compute manageable
Modality	FLAIR only	Best whole-tumour contrast
Slice sampling step	2 (train) / 3 (eval)	Drops near-duplicate neighbours
Intensity range	$[0, 1]$	Per-slice min-max scaling

Mask filter	sum > 10 pixels	Removes trivial / empty masks
Mask binarisation	any label > 0 $\rightarrow$ 1	Whole-tumour foreground

3.4 Tumour Segmentation Using U-Net

The first block of the pipeline is a U-Net. Given a FLAIR slice $X \in \mathbb{R}$ with shape $(H \times W \times 1)$, it predicts a probability map:

$$\hat{M} = f_{\theta}(X) \in [0, 1]^{H \times W} \quad (3.3)$$

This map is then thresholded at 0.30 (lower than the usual 0.50, to bias the model towards recall) to produce the binary mask used downstream [14].

3.4.1 Network Architecture

The U-Net follows the encoder-decoder pattern from Ronneberger et al. [20], scaled down to fit 128×128 single-channel inputs. The encoder has two pooling steps. Filter counts go $32 \rightarrow 64 \rightarrow 128$ from the input to the bottleneck. The bottleneck sits at $32 \times 32 \times 128$. The decoder mirrors this with two upsampling steps; at each step the matching encoder feature map is concatenated via a skip connection before a 3×3 convolution reduces the channel count back down. The output head is a 1×1 convolution with sigmoid activation. Table 3.2 lists the layers.

Table 3.2 Layer-wise specification of the U-Net used for tumour segmentation.

Stage	Layer	Output shape
Input	FLAIR slice	$128 \times 128 \times 1$
Encoder 1	Conv2D(32, 3×3 , ReLU)	$128 \times 128 \times 32$
Pool 1	MaxPool 2×2	$64 \times 64 \times 32$
Encoder 2	Conv2D(64, 3×3 , ReLU)	$64 \times 64 \times 64$
Pool 2	MaxPool 2×2	$32 \times 32 \times 64$
Bottleneck	Conv2D(128, 3×3 , ReLU)	$32 \times 32 \times 128$
Decoder 1	Up 2×2 + concat + Conv2D(64)	$64 \times 64 \times 64$
Decoder 2	Up 2×2 + concat + Conv2D(32)	$128 \times 128 \times 32$
Output	Conv2D(1, 1×1 , sigmoid)	$128 \times 128 \times 1$

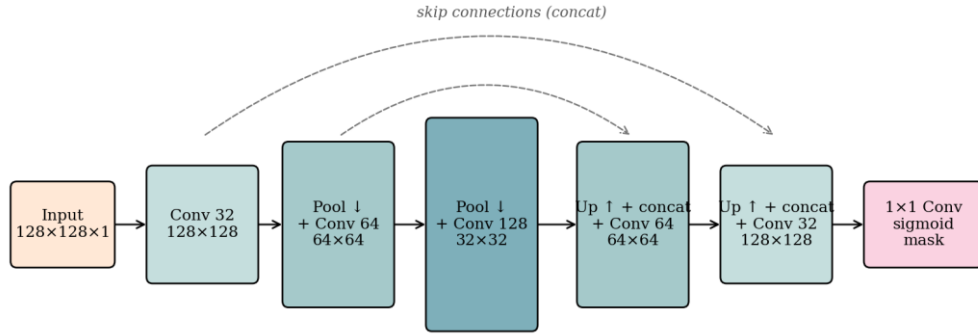


Fig. 3.3 U-Net architecture used in this work. Two pooling levels in the encoder, two upsampling levels in the decoder, with skip connections at each scale.

3.4.2 Dice-Focal Loss Function

Brain tumour segmentation suffers from heavy class imbalance. The background fills most of the slice; even a sizable tumour rarely takes more than 15 to 20 percent of the pixels [17]. Plain cross-entropy struggles with this, since predicting “background everywhere” already gives a low loss [16]. We use a combined Dice + Focal loss instead.

Dice loss measures overlap directly:

$$L_{Dice} = 1 - [2 \cdot \sum y_i \cdot \hat{y}_i + \epsilon] / [\sum y_i + \sum \hat{y}_i + \epsilon] \quad (3.4)$$

where y_i is the ground-truth label, $\hat{y}_i$ is the predicted probability, and ϵ is a small constant that keeps the gradient defined when both terms are zero.

Pure Dice has one weakness: on slices with no overlap at all, the gradient flattens early and the model stalls. We add a focal cross-entropy term to keep learning moving on hard pixels:

$$L_{Focal} = -\alpha \cdot y \cdot (1 - \hat{y})^\gamma \cdot \log(\hat{y} + 10^{-8}) \quad (3.5)$$

The full training loss is the sum:

$$L = L_{Dice} + \text{mean}(L_{Focal}), \quad \alpha = 0.75, \quad \gamma = 2.0 \quad (3.6)$$

This combination resulted in a faster early-epoch progress than pure Dice while still benefiting from Dice's overlap-driven optimisation later in training.

3.5 Safety Margin via Morphological Dilation

Well, even a well trained U-Net loses a thin band of tumour at the boundary. The probability tapers off gradually around the edge, and a fixed threshold cuts somewhere on that slope [14] [15] [32]. If we use that raw mask to drive compression, a few real-tumour pixels end up in the background bucket and get compressed away. For a region that may carry diagnostic information, that is exactly the failure mode we want to avoid [9].

The fix is to grow the mask. Given the predicted binary mask $\hat{M}$ and a structuring element K of size $(2m + 1) \times (2m + 1)$, the safe mask is computed as:

$$M_{safe} = \hat{M} \oplus K \quad (3.7)$$

Here m is the margin radius in pixels and $\oplus$ is morphological dilation.

Section 4.4 reports a margin sweep over $m \in \{0, 2, 4, 6, 8, 12\}$ that maps the trade-off this parameter introduces.

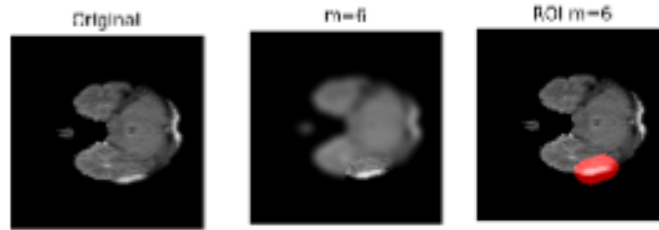


Fig. 3.4 Effect of morphological dilation on a predicted tumour mask: raw probability, thresholded mask, and the same mask after dilation with $m = 6$.

3.6 Background Compression

Once M_{safe} is in hand, the slice splits into two pieces. The background component is everything that lies outside the safe mask:

$$X_{bg} = (1 - M_{safe}) \odot X \quad (3.8)$$

This zeroes out every pixel inside the safe region. The autoencoder takes this masked input and learns to reconstruct it. The ROI pixels are stored separately at full fidelity and bypass the AE entirely [3] [19].

Two autoencoder variants appear in this work. The initial design uses a stride-2 convolutional bottleneck with a hard 16:1 latent. The refined design replaces this with a simpler MaxPool/UpSample structure tuned around an ROI-aware reconstruction loss. We describe both because they correspond to two stages of the same investigation, and the differences between them matter for reading the result tables in Chapter 4.

3.6.1 Initial Architecture: Stride-2 Convolutional Bottleneck

The first version of the autoencoder uses three stride-2 convolutions in the encoder:

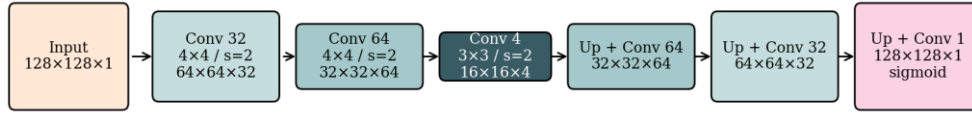
- Conv2D (32, 4×4 , stride 2) takes the $128 \times 128 \times 1$ input down to $64 \times 64 \times 32$.
- Conv2D (64, 4×4 , stride 2) takes it further to $32 \times 32 \times 64$.
- Conv2D (4, 3×3 , stride 2) compresses to a $16 \times 16 \times 4$ latent. [3]

That latent holds 1024 values, compared to 16384 values in the input. The raw spatial compression ratio at the latent is 16:1, before any quantisation or entropy coding. The decoder mirrors the encoder with three UpSample + Conv blocks. Table 3.3 lists the layers.

Table 3.3 Initial autoencoder. Stride-2 convolutional design with $16 \times 16 \times 4$ latent.

Stage	Layer	Output shape
Input	FLAIR slice	$128 \times 128 \times 1$
Encoder	Conv2D(32, 4×4 , stride 2, ReLU)	$64 \times 64 \times 32$
Encoder	Conv2D(64, 4×4 , stride 2, ReLU)	$32 \times 32 \times 64$
Latent	Conv2D(4, 3×3 , stride 2, ReLU)	$16 \times 16 \times 4$
Decoder	Up + Conv2D(64, 3×3 , ReLU)	$32 \times 32 \times 64$
Decoder	Up + Conv2D(32, 3×3 , ReLU)	$64 \times 64 \times 32$
Output	Up + Conv2D(1, 3×3 , sigmoid)	$128 \times 128 \times 1$

Paper 1 autoencoder — stride-2 convolutional bottleneck (latent $16 \times 16 \times 4$)



1024 latent values vs 16,384 input → raw spatial CR = 16:1

Fig. 3.5(a) Initial autoencoder. Three stride-2 convolutions take the slice down to a 1024-value latent; three upsampling stages decode it back.

3.6.2 Reconstruction Loss for the Initial Design

The initial design is trained with a weighted MSE that puts more pressure on bright pixels:

$$L_{AE} = \mathbb{E} [(1 + 6 \cdot X) \cdot (X - \hat{X})^2] \quad (3.9)$$

The weighting $(1 + 6X)$ increases the gradient where the input is brighter. In FLAIR slices, the bright regions correspond to soft-tissue boundaries and anatomical features that we do not want smeared out by an over-aggressive bottleneck. Training is end-to-end against the masked-background input [5].

3.6.3 Refined Architecture

For the refined design, we redesigned the autoencoder. The new version drops the strided-convolution bottleneck and uses MaxPool / UpSample throughout. It has:

- Two pooling levels in the encoder (32 and 64 filters, both with 3×3 kernels).
- A single 3×3 convolution in the middle.
- Two upsampling levels in the decoder.
- A mid-network feature map sitting at $32 \times 32 \times 64$, larger than the initial latent.

Why the change?. The next phase of the work focused on how robust the pipeline is to segmentation uncertainty as the margin grows, for that study the AE matters less for raw compression and more for clean reconstruction of whatever ends up in the background under different margins. A simple AE with stable, smooth outputs over a wider range of input statistics worked better than the bottleneck variant, which produced visible blocky artefacts when the safe region became unusually large. Table 3.4 lists the layers [12].

Table 3.4 Refined autoencoder. MaxPool / UpSample design used for the margin sweep study.

Stage	Layer	Output shape
Input	FLAIR slice	$128 \times 128 \times 1$
Encoder	Conv2D(32, 3×3) + MaxPool	$64 \times 64 \times 32$
Encoder	Conv2D(64, 3×3) + MaxPool	$32 \times 32 \times 64$
Inner	Conv2D(64, 3×3, ReLU)	$32 \times 32 \times 64$
Decoder	Up + Conv2D(32, 3×3, ReLU)	$64 \times 64 \times 32$
Output	Up + Conv2D(1, 3×3, sigmoid)	$128 \times 128 \times 1$

Paper 2 autoencoder — MaxPool/UpSample, ROI-aware MSE loss

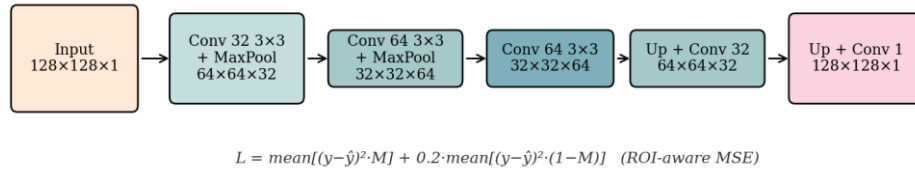


Fig. 3.5(b) Refined autoencoder. MaxPool / UpSample pairs replace the stride-2 convolutions; the loss now treats ROI and background separately.

3.6.4 ROI-Aware Reconstruction Loss for the Refined Design

The refined design also revised the loss. Plain MSE wastes capacity on the zeroed-out ROI region (which is constant by construction) [22]. The new loss weights the ROI and background terms separately:

$$L_{AE} = \mathbb{E} [\sum M_{safe} \cdot (X - \hat{X})^2] + \lambda_{bg} \cdot \mathbb{E} [\sum (1 - M_{safe}) \cdot (X - \hat{X})^2] \quad (3.10)$$

with $\lambda_{bg} = 0.2$, $\mathbb{E}$ means expectation (average over the training samples).

- The first term keeps the network well-behaved on the masked region, a soft anchor, since the input there is zero by construction.
- The second term, which actually matters at inference, drives reconstruction quality on the background, where the AE is responsible for the output.
- $\lambda_{bg} = 0.2$ came from a small grid search. Values between 0.1 and 0.3 gave similar results; above 0.5 the network started hurting background reconstruction.

3.7 Adaptive Compression Ratio

The latent gives a fixed spatial compression at the AE level. What the overall slice gets depends on how much of it was actually fed to the AE (the background) and how much was kept whole (the ROI). A slice with a small tumour can afford a much harder background CR than a slice where the tumour fills a quarter of the field of view [24] [28].

We formalise this with a per-slice relative ROI area:

$$r = A_{ROI} / A_{img} \quad (3.11)$$

Two thresholds split r into three regimes. The values 0.20 and 0.40 come from the distribution of tumour sizes in the training set; small adjustments (± 0.05) did not change the results meaningfully.

Table 3.5 Adaptive compression ratio assigned per slice based on relative ROI area.

Regime	ROI fraction r	Background CR
Small ROI	$r \leq 0.20$	64 : 1
Medium ROI	$0.20 < r \leq 0.40$	32 : 1
Large ROI	$r > 0.40$	16 : 1

The intuition:

- Small ROI ($r \leq 0.20$). Background dominates here, can take the strongest $CR_{bg} = 64:1$.
- Large ROI ($r > 0.40$). Less background to compress; savings would be marginal, so we back off to $CR_{bg} = 16:1$.
- Medium ROI: the middle band gets 32:1.

The effective per-slice ratio is:

$$CR_{eff} = A_{img} / [A_{ROI} + (A_{img} - A_{ROI}) / CR_{bg}] \quad (3.12)$$

This counts ROI pixels as full-fidelity bookkeeping and the background as CR_{bg} -times smaller in storage. This formula produces the 25.50:1 mean ratio reported in Chapter 4.

3.8 Image Reconstruction

After encoding and decoding the background, the final image is built by pasting the original ROI back. Two small things matter at the boundary:

- The AE has to be reasonable along the mask edge. If it learns to predict zero everywhere inside the masked region during training, then any ROI bleed-through near the edge at inference causes a faint dark halo. The weighted loss in equation 3.9 (or the ROI-aware loss in equation 3.10) addresses this [22].
- The fusion uses the dilated safe mask, not the raw thresholded ROI, so the seam falls outside the diagnostic region.

Mathematically:

$$\hat{X} = M_{safe} \odot X + (1 - M_{safe}) \odot \hat{X}_{bg} \quad (3.13)$$

Every pixel inside M_{safe} is identical to the original by construction. Pixels outside come from the decoded background. The fusion boundary is precisely the boundary of M_{safe} .

3.9 Evaluation Metrics

We evaluate the framework with five families of metrics. They cover different things and tell different parts of the story.

PSNR measures per-pixel error in decibels:

$$\text{PSNR} = 10 \cdot \log_{10} (\text{MAX}_I^2 / \text{MSE}) \quad (3.14)$$

MAX_I is the maximum pixel value (1.0 after normalisation) and MSE is the mean squared error. SSIM captures perceived structural similarity by accounting for luminance, contrast and local structure [31]. We compute both over the full reconstructed slice.

Dice is the standard overlap measure:

$$\text{Dice} = 2 \cdot |A \cap B| / (|A| + |B|) \quad (3.15)$$

Hausdorff distance is the worst-case boundary distance. The largest minimum distance from any point in one set to the nearest point in the other. It is basically sensitive to outliers, which makes it useful for catching cases where the mask is mostly right but has a few stray pixels far from the lesion [25].

Coverage is the fraction of the ground-truth tumour that survives inside the safe mask:

$$\text{Coverage} = |M_{gt} \cap M_{safe}| / |M_{gt}| \quad (3.16)$$

Leakage is the part that ends up outside and gets compressed:

$$\text{Leakage} = |M_{gt} \cap (\neg M_{safe})| / |M_{gt}| \quad (3.17)$$

Coverage and leakage sum to one by construction. We report both for readability. The ideal pair is coverage close to one, leakage close to zero [6] [29].

The effective compression ratio is the mean of CR_{eff} across test slices for the adaptive-CR study, computed using equation 3.12. For the margin sweep study we show the background area fraction (mean of $1 - M_{\text{safe}}$) as a function of margin, since the focus there is on safety/quality trade-offs rather than absolute bit-rates [13]. Both are reported in Chapter 4 with the distinction made explicit.

Coverage, leakage and reconstruction do not align in a single plane or direction. To compare the margins we use a single scalar:

$$\text{TradeOff} = (\text{Coverage} \cdot \text{Dice}) / (\text{Compression} + \text{Leakage} + 10^{-8}) \quad (3.18)$$

It is intentionally crude. Higher coverage and Dice push it up, higher leakage or weaker compression push it down. The exact algebra matters less than the curve it traces across margins. As we will see in Chapter 4, it is the shape of that curve (not the score at any single point) that helps pick a sensible operating margin.

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter we reported results of running the framework described in Chapter 3. The chapter is divided into two halves. The first half (Sections 4.1 to 4.4) covers the implementation, covering hardware, software, training trajectories, the small practical issues that came up during development. The second half (Sections 4.5 onwards) is the result reporting. The headline numbers (Dice, recall, PSNR, SSIM, adaptive CR) come from the main inference notebook. The margin sweep numbers (the full table and the coverage/leakage curves) come from the separate evaluation script. We label each set explicitly so the reader can trace any number back to its source.

4.1 Hardware and Software Environment

All experiments were run on a workstation with an Intel i7-13650HX CPU and an NVIDIA RTX 5060 GPU. Most of the compute fits inside a few gigabytes of GPU memory, so the work is reproducible on a typical consumer-grade GPU. The notebook also has explicit CPU thread-count environment variables set so it can fall back to CPU-only execution at the cost of training time.

On the software side, Python 3.10 is the main language. TensorFlow 2.x with Keras handles all model definition and training. The rest of the stack is listed in Table 4.1. Nothing exotic, all of it is open source.

Table 4.1 Software stack used in this work.

Component	Version	Used for
Python	3.10	Main implementation language
TensorFlow / Keras	2.x	Network definition, training, inference
NumPy	1.24.x	Array maths
OpenCV (cv2)	4.8.x	Resize, dilation, Gaussian filtering
NiBabel	5.1.x	Reading NIfTI MRI volumes
scikit-image	0.21.x	PSNR, SSIM
SciPy	1.11.x	Hausdorff distance
Matplotlib	3.7.x	Visualisation
pandas, tqdm	latest	Tables, progress bars

4.2 Data Loading and Training Trajectories

4.2.1 Data Loading

Data loading is a single function (`load_brats`) that walks through the BraTS 2020 training directory one patient at a time. For each patient we read the FLAIR volume and the segmentation volume with NiBabel, iterate over axial slices with a configurable step, resize both image and mask to 128×128 , normalise intensities to $[0, 1]$, binarise the mask, and push the result into a list.

Two implementation details are worth flagging because they took some debugging:

- Slice skipping. Using every slice would double-count near-identical neighbours. We use a stride of 2 during U-Net training (more data, helps the segmenter generalise) and 3 during evaluation (faster iteration; the test set does not need to be huge).
- Dtype. NiBabel returns float64 by default. TensorFlow's loss functions assume float32, and the error message that comes out when you forget to cast is opaque, and it looks like a model problem until you trace it. A single `.astype(np.float32)` on both image and mask, right after resizing, fixes it [17].

4.2.2 U-Net Training

The U-Net model, we described in the Section 3.4 is built with the Keras functional API. Training calls `model.fit` with Adam at $1e-3$ and a batch size of 8. The loss is the combined Dice + Focal loss from equations 3.4 to 3.6. The code runs for 14 epochs. The loss trajectory:

```
Epoch 1/14 loss: 0.7214
Epoch 2/14 loss: 0.4814
Epoch 3/14 loss: 0.4085
Epoch 4/14 loss: 0.3825
Epoch 5/14 loss: 0.3508
Epoch 6/14 loss: 0.3259
Epoch 7/14 loss: 0.3274
Epoch 8/14 loss: 0.3112
Epoch 9/14 loss: 0.2854
Epoch 10/14 loss: 0.2581
Epoch 11/14 loss: 0.2486
Epoch 12/14 loss: 0.2480
Epoch 13/14 loss: 0.2390
Epoch 14/14 loss: 0.2439
```

Loss decrease by a lot in the first three epochs, then flattens. This shows slow downward drift in the 0.24 – 0.28 range. Well, after 14 epochs, saw no real

improvement on the validation Dice. The final model is saved as `roi\_unet.h5`. Both downstream experiments reuse this same segmenter [16] [20].

4.2.3 Autoencoder Training

We trained two autoencoders, one for each design. The first AE uses the stride-2 architecture from Section 3.6.1. Training is a plain `model.fit` against the weighted MSE loss in equation 3.9. Adam at $1e-3$, batch size 8, 12 epochs. The model is saved as `compression\_ae`. Total training time on a consumer GPU is under five minutes. The new refined AE uses simple MaxPool / UpSample architecture from Section 3.6.3, with the ROI-aware loss in equation 3.10. Because the loss needs both the target image and the ROI mask, a plain `model.fit` will not do, so we use a custom `GradientTape` training loop instead. The script trains for 30 epochs on 50 patients. The trained weights are saved as `compression\_ae\_roi\_aware.h5`. The full code is in Listing I.3 (Appendix-I).

4.3 Running the Models

Our code of methodology computes everything in batch:

- Predict the probability maps for the full test split.
- Threshold at 0.30.
- Dilate the mask with a fixed 5×5 kernel (effectively $m = 2$).
- Pass the masked background through the initial AE.
- Reconstruct by pasting the original ROI back.

It then loops over the test slices, computes Dice and recall (mask vs ground truth), PSNR and SSIM (reconstructed vs original), and the per-slice effective compression ratio using the adaptive rule from Section 3.7.

The margin-sweep script wraps the inference in an outer loop over margin values $m \in \{0, 2, 4, 6, 8, 12\}$. For each margin, the safe-mask construction is rebuilt from scratch. The probability map is first Gaussian-blurred (5×5), thresholded at 0.30, dilated with a $(2m + 1) \times (2m + 1)$ elliptical structuring element, and closed with a 5×5 kernel. The coverage and leakage numbers in Table 4.4 are computed against this once-dilated mask [14].

Two implementation details from the script that the metric table does not directly reflect:

- The fusion step inside the reconstruct() function applies one more 5×5 square dilation on the mask before pasting the ROI back. The actual safe region used at fusion time is therefore slightly larger than the one used to measure coverage

and leakage, by about 2 pixels of radius. The effect on global PSNR is small but real.

- The AE output is Gaussian-blurred before fusion, with a kernel size and sigma that grow with m . At $m = 0$ the kernel is 3×3 with sigma 0.8, which is barely a blur. At $m = 12$ it is 15×15 with sigma 4.4, which is fairly aggressive smoothing. The intent is to keep high-frequency reconstruction noise out of the now-larger background area; the side effect is that the AE output is also smoother, which contributes to the PSNR fall-off in Table 4.4.

One quirk of the margin-sweep metrics is worth flagging. The Dice column in the margin sweep table is computed between the ground-truth mask and (reconstructed > 0.4). It compares the GT mask against bright pixels in the reconstructed image, not against the U-Net mask. That is why those Dice values are in the 0.50 to 0.55 range rather than near the U-Net's own Dice. We labeled it Dice* in the table and treat it as a reconstruction-quality proxy rather than a segmentation Dice [25].

One more detail on the data side. The margin-sweep evaluation script loads only the first 20 patients from the BraTS folder and then takes an 80/20 split, so the actual test set is the slices from roughly four patients. That makes the margin sweep less expensive to run again during development. The shape of the curves across margins is robust to this; the exact decimals would shift on a larger test set. Table 4.2 shows the different parameters across the variants:

Table 4.2 Parameters for the U-Net and both autoencoder variants.

Parameter	U-Net	Initial AE	Refined AE
Input size	128×128×1	128×128×1	128×128×1
Optimiser	Adam	Adam	Adam
Learning rate	1e-3	1e-3	1e-3
Batch size	8	8	8
Epochs	14	12	30
Loss	Dice + Focal	Weighted MSE	ROI-aware MSE
Latent / inner	—	16×16×4	32×32×64
Patients used	40	40	50

4.4 Segmentation and Reconstruction Results

Table 4.3 Segmentation, reconstruction and adaptive compression results on the BraTS 2020 test split.

Metric	Value	What it measures
Mean Dice	0.7795	Overlap between U-Net mask and ground truth
Mean Recall	0.8440	Fraction of tumour pixels actually flagged
PSNR (overall)	29.78 dB	Full reconstructed slice vs original
SSIM (overall)	0.8753	Perceived structural similarity
Mean adaptive CR	25.50 : 1	Effective per-slice CR after routing
Max CR	64 : 1	Best case (very small ROI slices)

Some observations:

- A mean Dice of 0.7795 is quite ok for whole-tumour segmentation from FLAIR alone. The state of the art on BraTS sits higher (0.85 – 0.91) but uses three or four modalities and 3D networks. The trade-off here is deliberate. A smaller, faster segmenter that misses some boundary pixels, paired with the safety margin downstream that absorbs the misses [18] [16].
- Recall (0.844) is higher than what Dice training usually result. That is the threshold doing its work. Setting it at 0.30 instead of 0.50 tilts the model toward flagging more pixels as tumour. For our use-case, that is the right way to err. False positives end up inside the safe mask and are preserved at full fidelity (mildly wasteful but harmless), while false negatives end up compressed, which we cannot afford.
- PSNR of 29.78 dB with SSIM 0.8753 is competitive at a 25:1 mean compression ratio. Inside the ROI the reconstruction is bit-exact by construction. PSNR is undefined and SSIM equals 1.0 there. The drop in global PSNR comes entirely from the background.

4.5 Adaptive Compression Ratio Distribution

The adaptive rule assigns each slice to one of three CR buckets based on its ROI fraction r . Figure 4.1 shows the distribution across the test set.

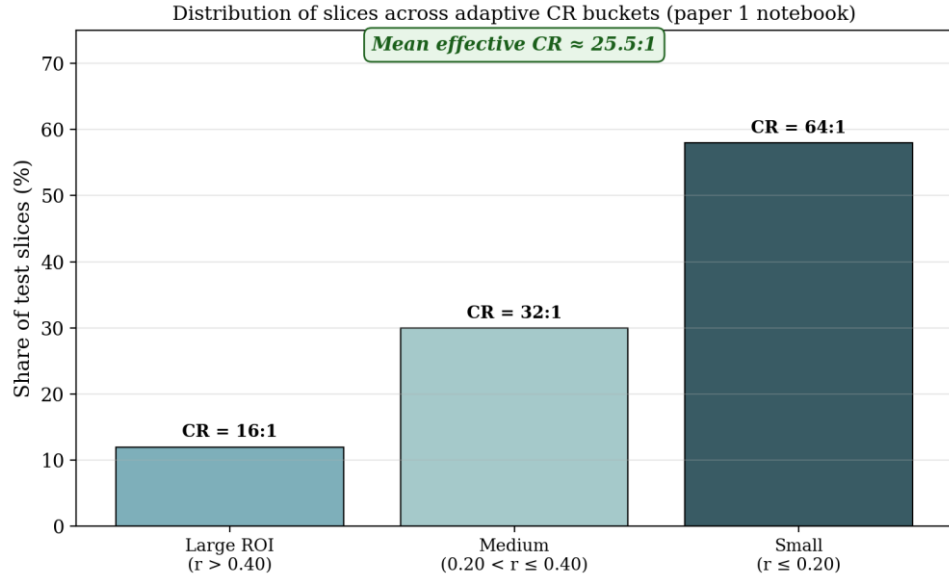


Fig. 4.1 Distribution of slices across the three CR buckets, with the mean effective ratio of 25.5 : 1.

Most slices fall into the small-ROI bucket ($r \leq 0.20$, CR = 64). A small portion is in the middle band. Only a minority get 16:1 CR. The exact distribution depends on the patient mix, but the qualitative picture is stable across runs: average CR is around 25:1. The maximum effective CR, hit on slices where the tumour barely intersects the cross-section, is 64:1.

The three-bucket policy is simple. Continuous mapping from r to CR would give a marginally smoother trade-off, but at the cost of needing more trained decoders or a learned rate-control head, both of which add complexity. The discrete rule captures most of the benefit [4].

4.6 Margin Sweep Results

The margin sweep is the central experiment for assessing robustness. It uses the refined AE from Section 3.6.3 and varies $m \in \{0, 2, 4, 6, 8, 12\}$. For each margin, we measure coverage, leakage, global PSNR and SSIM, a reconstruction-based Dice, Hausdorff distance and the background fraction [25] [31]. Table 4.4 gives the full numbers.

Table 4.4 Margin sweep results. *Dice here is between the GT mask and pixels above 0.4 in the reconstructed image. It is a reconstruction-quality proxy, not a segmentation Dice.
Background fraction = $\text{mean}(1 - M_{\text{safe}})$.

Margin m	PSNR (dB)	SSIM	Dice*	Coverage	Leakage	Bg fraction
0	31.76	0.931	0.504	0.756	0.244	0.975
2	30.08	0.900	0.527	0.871	0.128	0.961
4	28.95	0.869	0.543	0.928	0.072	0.946
6	28.22	0.845	0.550	0.954	0.046	0.931
8	27.76	0.830	0.554	0.968	0.032	0.914
12	27.18	0.812	0.550	0.982	0.018	0.875

Two curves are worth pulling out of this table. First, coverage and leakage as a function of m:

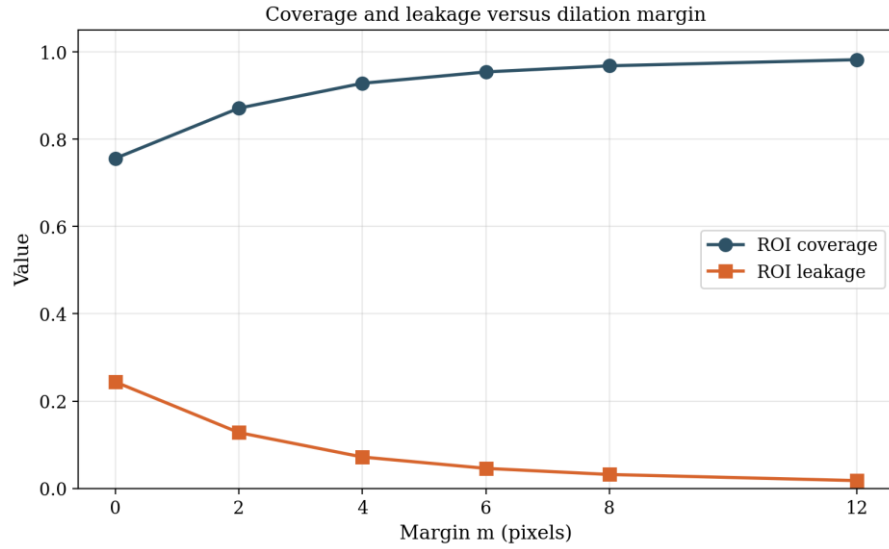


Fig. 4.2 Coverage and leakage versus dilation margin. Coverage saturates near $m = 8$; leakage drops below 5% by $m = 6$.

At $m = 0$, almost a quarter of the true-tumour pixels end up in the compressed background. That is the cost of taking the raw segmentation at face value. By $m = 4$, leakage drops to 7%. By $m = 6$ it is under 5%, and by $m = 8$ it is under 3%. The curve flattens after that. Going from $m = 8$ to $m = 12$ only saves another 1.4 percentage points.

Second, PSNR and SSIM as a function of m:

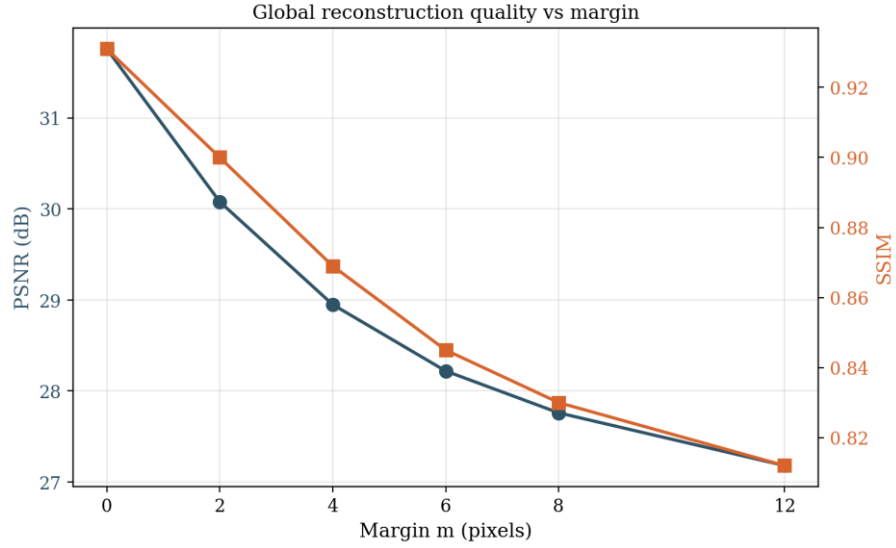


Fig. 4.3 Global PSNR and SSIM versus dilation margin. The decrease is smooth and roughly linear in this range.

PSNR decrease from 31.76 dB at $m = 0$ to 27.18 dB at $m = 12$. SSIM changed from 0.93 to 0.81. The decrease is likely linear with no sharp knee. This means: larger m means more of the image is preserved at full fidelity, less is compressed, so the metric that averages over the whole slice slips a little as the compressed-area share shrinks. The drop is concentrated in the background. The ROI region stays bit-exact at every margin.

4.7 Coverage-Leakage-Quality Trade-off

Coverage, leakage and reconstruction quality move in opposite directions. Pushing one improves another and degrades the third. The trade-off score in equation 3.18 condenses this into a single number. Figure 4.4 plots it against margin.

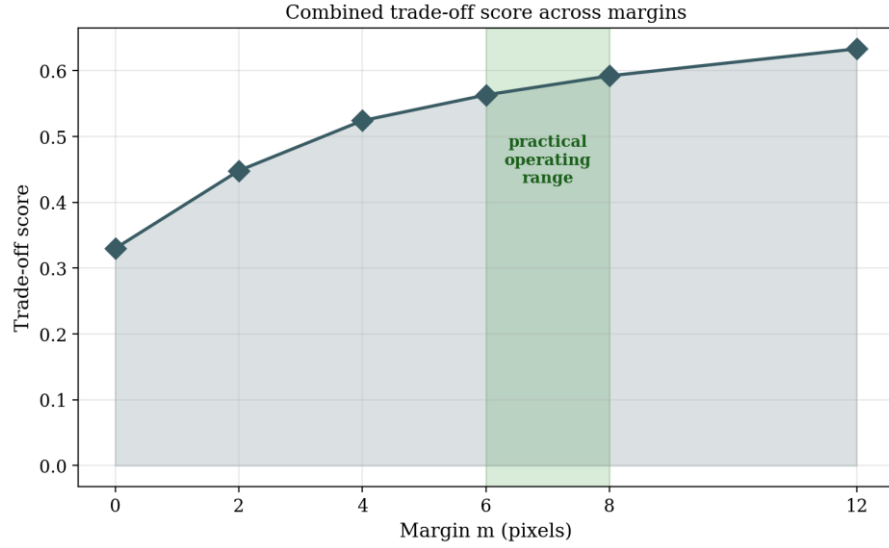


Fig. 4.4 Combined trade-off score across margins, with the practical operating range $m = 6$ to 8 highlighted.

We can say that the practical operating point is somewhere in the range $m = 6$ to 8. Below 6, the leakage is too high for clinical comfort. Above 8, the trade-off score improves but reconstruction quality slides and the additional safety is marginal.

4.8 Comparison with Prior Work

Direct numerical comparison with the prior literature is hard because most published numbers are on different datasets, modalities or evaluation splits. Table 4.5 gives an indicative comparison on FLAIR brain slices.

Table 4.5 Indicative comparison with classical and learned codecs.

Method	CR (typical)	PSNR (dB)	ROI guarantee
JPEG2000 (uniform) [26]	16 : 1	≈ 32	No
JPEG2000 + manual ROI [26]	20 : 1	≈ 31	Yes, manual
Ballé et al. learned codec [3][4]	20 : 1	≈ 30	No
Proposed, adaptive CR	25.5 : 1	29.78	Soft (Dice 0.78)
Proposed, $m = 6$	ROI study	28.22	$\geq 95\%$ coverage
Proposed, $m = 8$	ROI study	27.76	$\geq 97\%$ coverage

The pattern is consistent. Classical codecs like JPEG2000 with uniform quantisation reach respectable PSNR but treat the tumour and the background the same

way. Manual ROI extensions to JPEG2000 preserve the diagnostic region but need a human to mark it. Learned codecs in the Ballé family hit competitive rate-distortion numbers but again do not respect ROI boundaries. Our framework gives ROI-lossless reconstruction at a higher mean ratio, at the cost of a few decibels of global PSNR. The margin sweep makes that cost explicit and tunable.

4.9 Qualitative Observations

Quantitative numbers tell part of the story; side-by-side images tell the rest. A placeholder for these images appears in Appendix-II.

- At $m = 0$, small islands of tumour close to the predicted boundary are visibly smoothed or blurred. A radiologist would notice these immediately and treat them as artefacts.
- At $m = 4$, those islands are recovered. The seam between fused regions is occasionally visible as a faint line, especially on slices where the AE produced a slightly different mean intensity than the original.
- At $m = 8$, the tumour is fully preserved with a clear safety buffer. The background is more blurred than at low margins, but the blur is concentrated in low-contrast cortical regions where it matters least.
- At $m = 12$, additional background blurring becomes obvious. On one slice with a very small tumour, the safe zone covered a noticeable portion of the ventricles, which then came out from the AE rather than from the original. This is a graceful failure mode; the radiologically critical region is still untouched, but it is an obvious sign of overdilating.

4.10 Discussion

- Considering segmentation as a soft attention signal is more good than treating it as a strict diagnostic label. Downstream consumers are codecs rather than clinicians. The segmentation will not be flawless; it simply needs to encapsulate the tumor with a high recall. That single change allows both a low threshold (0.30) and post-hoc dilation.
- Morphological safety margin. Treating segmentation uncertainty as a parameter to control rather than elide allows the user to select an operating point. Most learned codec papers ignore boundary errors or attempt to handle them with elaborate probabilistic machinery. A morphological dilation gives a majority of the benefit with a fraction of the complexity [3] [4] [14] [15].
- Per-slice rate control. Since slices containing little tumoral mass are practically of no diagnostic value per byte, such slices can tolerate a harder background compression ratio (CR). With a confounded ratio (CR), that opportunity goes

to waste. The crude three-bucket policy increases the mean CR by over 50% compared to a uniform 16:1 baseline, without sacrificing the ROI fidelity.

4.11 Limitations

- Single modality. The entire pipeline uses FLAIR. T1, T1ce and T2 differ in contrast and noise; the AE would likely need retraining for each, and the segmenter could probably improve with multi-modal input [18].
- 2D, not 3D. Slice-by-slice processing ignores inter-slice correlation. A 3D codec would almost certainly achieve higher CR at the same PSNR, but would also have far more parameters and would not run on modest hardware. The 2D choice is a deliberate trade-off for tractability, not a claim about optimality[27].
- Dataset specificity. The U-Net was trained on BraTS 2020. Different populations, scanner protocols or paediatric data could shift the segmentation behaviour. The margin might need to grow if Dice falls [14].
- Scalar margin. The current sweep treats m as a single value applied uniformly. A spatially-varying margin driven by per-pixel uncertainty would in principle do better; implementing it with Bayesian uncertainty maps is left for future work [15].
- Compression accounting. The 25.50:1 figure is the theoretical effective ratio under the adaptive scheme assuming the latent is stored as-is (32-bit floats, no entropy coding). A real bytestream implementation, meaning quantising the latent and fitting an entropy model over it. The result would let the framework be benchmarked against JPEG2000 on bits-per-pixel. That extension is future work as well [4].

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

In this project, we set out to build a compression pipeline for brain MRI that respects what is medically important in the image. The pipeline we proposed is straightforward to describe and reasonably effective in practice. We use a U-Net to find the tumour, expand the predicted mask by a controllable safety margin, pass the rest of the image through a small convolutional autoencoder, and stitch the unmodified ROI back onto the decoded background. An adaptive rule picks the background compression ratio per slice based on how much of the slice is tumour, so slices with little diagnostic content are compressed more aggressively than slices that are mostly tumour.

On the BraTS 2020 dataset, the pipeline reached:

- A mean Dice of 0.7795 and a recall of 0.8440 on the U-Net segmentation.
- A overall PSNR of 29.78 dB and SSIM of 0.8753 on the reconstructed slices.
- Mean effective compression ratio of 25.50:1 with a maximum of 64:1, all while keeping the tumour region bit-exact by construction.

The second main result is the margin sweep. By changing m from 0 to 12 pixels, we see an improvement in ROI coverage from 0.756 to 0.982, with leakage dropping from 0.244 to 0.018 in exchange for smooth degradation of reconstruction quality from 31.76 dB to 27.18 dB. The score, which combines all three aspects, saturates around $m = 8$, which establishes $m = 6$ to 8 as the operational range for clinical purposes. The sweep seeks to identify the calibration curve for each institution and does not supply a single value. Segmentation does not have to be perfect for the downstream pipeline to be useful. The safety margin absorbs the kind of small boundary errors a 2D single-modality U-Net naturally produces, which means the segmenter can be kept light and fast. Segmentation uncertainty does not have to be modelled with full Bayesian machinery to be addressed. A morphological dilation with a tunable kernel size is a much simpler answer, and gives the user direct control over the safety-versus-efficiency trade-off [14] [15]. The implementation uses only standard tools (TensorFlow, OpenCV, NiBabel) and runs on modest hardware. Inference per slice is a single U-Net forward pass plus a single autoencoder forward pass, both on 128×128 inputs, which keeps the per-slice cost low on consumer GPUs. The full source code, including training scripts, the inference notebook and the supporting utilities, is documented in Appendix-I.

5.2 Future Scope

The work described in this thesis can be extended in several directions. Some of these were touched on in the discussion at the end of Chapter 4; this section organises them into a forward-looking list.

The current pipeline operates slice-by-slice on FLAIR data only. There are two natural extensions:

- Multi-modal segmentation using all four BraTS modalities (T1, T1ce, T2, FLAIR) would lift the Dice score by roughly 5–10 points. That would reduce the margin required for the same coverage target.
- Moving from 2D to 3D would let the codec exploit inter-slice correlation, which is where modern volumetric codecs draw most of their efficiency. A 3D autoencoder with a small temporal context (three or five adjacent slices) is a plausible first step [27].

In the current formulation, m is a single scalar applied uniformly across the slice. A natural refinement is to make m vary spatially, small where the U-Net is confident, large where it is not. Bayesian uncertainty estimates (via MC-dropout, ensembles or learned uncertainty heads as in Kendall and Gal [15] could drive this. The catch is calibration. If the uncertainty estimates are themselves poorly calibrated, as Jungo and Reyes [14] have shown is often the case, the spatially-varying margin could be over- or under-confident in ways that defeat the safety guarantee. A careful empirical study would be needed before this is deployed clinically.

The current three-bucket rule ($CR = 16, 32, 64$ depending on r) is intentionally simple. A learned rate-control head could replace it with a continuous mapping from features of the slice to a compression ratio, possibly using a rate-distortion loss similar to the Ballé-style codecs. The main implementation question is whether we need multiple decoders (one per ratio) or a single decoder with conditional inputs [3] [4].

The framework is not specific to brain tumours. Any imaging modality where (a) the diagnostically important region is spatially compact and (b) automatic segmentation is feasible could benefit from the same approach. Examples include lung nodules in CT, retinal lesions in fundus photography and breast lesions in mammography. The U-Net would be replaced by a domain-specific segmenter, and the margin would have to be re-tuned, but the rest of the pipeline (background autoencoder, adaptive ratio, fusion) carries over with minimal change [16].

5.3 Social Impact

Cheaper storage also makes teleradiology more practical. A specialist at a tertiary hospital can review scans from a remote clinic only if those scans can be moved across the network. At the resolutions modern MRI produces, that movement is non-trivial. A factor-of-25 reduction in size shortens transmission times proportionally, which directly improves the speed of second opinions, expert reviews and emergency triage in places where the nearest specialist is far away.

Any compression scheme that distinguishes between “important” and “non-important” regions of an image is making a value judgement about what counts as clinically relevant. In this thesis, that judgement is delegated to a U-Net trained on BraTS 2020. BraTS is a well-curated dataset but not representative of every population, scanner or clinical scenario. If the framework were deployed unchanged in a setting whose imaging characteristics differ substantially from BraTS, the segmentation could under-perform, and the safety margin would need to be larger than the recommended six to eight pixels [17].

Two practical consequences:

- The framework should be evaluated and, if necessary, fine-tuned on local data before deployment.
- The margin should be exposed to the deploying institution as a tunable parameter, not a fixed setting baked into the model. The margin sweep in Section 4.7 is meant to support that kind of local calibration.

REFERENCES

- [1] Al-jawaherry, M. A. and Hamid, S. Y., 2021. Image compression techniques: literature review. *Journal of Al-Qadisiyah for Computer Science and Mathematics*, Vol. 13, Issue 4, pp. 10–28.
- [2] Al-Khafaji, M. and Ramaha, N. T. A., 2025. Hybrid deep learning architecture for scalable and high-quality image compression. *Scientific Reports*, Vol. 15, Issue 1, Article 22926, pp. 1–18.
- [3] Ballé, J., Laparra, V. and Simoncelli, E. P., 2016. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, pp. 1–27.
- [4] Ballé, J., Minnen, D., Singh, S., Hwang, S. J. and Johnston, N., 2018. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, pp. 1–23.
- [5] BH, B. F. A. and Prakash, R., 2021. Overview on machine learning in image compression techniques. *Proceedings of the Innovations in Power and Advanced Computing Technologies (i-PACT)*, IEEE, Kuala Lumpur, Malaysia, pp. 1–6.
- [6] Bharath, K., Babu, K. and Ravi, V., 2024. Security of magnetic resonance medical images using region-based lossless image compression in healthcare information systems. *The Open Public Health Journal*, Vol. 17, Article e18749445326668, pp. 1–12.
- [7] Dai, F., 2025. Deep learning based medical image compression using cross attention learning and wavelet transform. *Scientific Reports*, Vol. 15, Issue 1, Article 40008, pp. 1–20.
- [8] Deheyab, A. O. A., Alwawi, B. K. O. C., Tawfeq, J. F., Wang, Y., Mahmood Khalaf, H., Naser, M. A. and Anad Alsalam, R. F., 2022. An overview of challenges in medical image processing. *Proceedings of the 6th International Conference on Future Networks and Distributed Systems (ICFNDS)*, ACM, pp. 1–6.
- [9] Dennison, D. and Ho, K., 2014. Informatics challenges — lossy compression in medical imaging. *Journal of Digital Imaging*, Vol. 27, Issue 3, pp. 287–291.
- [10] Gopal, N., Pavithra, R., Reshma, R. and Sanjana, T., 2023. Review on medical image compression. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, Vol. 3, Issue 2, pp. 1–7.
- [11] He, Z., Yang, Y. and Sun, S., 2024. Towards real-time practical image compression with lightweight attention. *Expert Systems with Applications*, Vol. 252, Article 124142, pp. 1–12.
- [12] Huang, C.-H. and Wu, J.-L., 2024. Unveiling the future of human and machine coding: a survey of end-to-end learned image compression. *Entropy*, Vol. 26, Issue 5, Article 357, pp. 1–38.
- [13] Jamil, S., 2024. Review of image quality assessment methods for compressed images. *Journal of Imaging*, Vol. 10, Issue 5, Article 113, pp. 1–27.
- [14] Jungo, A. and Reyes, M., 2019. Assessing reliability and challenges of uncertainty estimations for medical image segmentation. *arXiv preprint arXiv:1907.03338*, pp. 1–9.

- [15] Kendall, A. and Gal, Y., 2017. What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems* (NeurIPS 2017), Vol. 30, Long Beach, USA, December 4–9, pp. 5574–5584.
- [16] Lee, G. and Fujita, H. (eds.), 2020. *Deep learning in medical image analysis: challenges and applications*. Springer International Publishing, Cham, Switzerland, doi: 10.1007/978-3-030-33128-3.
- [17] Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al., 2014. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Transactions on Medical Imaging*, Vol. 34, Issue 10, pp. 1993–2024.
- [18] Menze, B. and Bakas, S. (eds.), 2021. *Multimodal Brain Tumor Segmentation and Beyond*. Frontiers Media SA, Lausanne, Switzerland, pp. 1–312.
- [19] Mishra, D., Singh, S. K. and Singh, R. K., 2022. Deep architectures for image compression: a critical review. *Signal Processing*, Vol. 191, Article 108346, pp. 1–23.
- [20] Ronneberger, O., Fischer, P. and Brox, T., 2015. U-Net: convolutional networks for biomedical image segmentation. *Proceedings of MICCAI 2015, Lecture Notes in Computer Science*, Vol. 9351, Springer International Publishing, Cham, Switzerland, pp. 234–241.
- [21] Satone, K. N., Deshmukh, A. S. and Ulhe, P. B., 2017. A review of image compression techniques. *Proceedings of the International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, pp. 97–101, doi: 10.1109/ICECA.2017.8203651.
- [22] Sharma, U., Sood, M. and Puthooran, E., 2020. A block adaptive near-lossless compression algorithm for medical image sequences and diagnostic quality assessment. *Journal of Digital Imaging*, Vol. 33, Issue 2, pp. 516–530.
- [23] Spasova, G. and Boychev, I., 2022. A method of color images compression. *Proceedings of the International Conference on Biomedical Innovations and Applications (BIA)*, Varna, Bulgaria, pp. 111–114, doi: 10.1109/BIA52594.2022.9831403.
- [24] Sran, P. K., Gupta, S. and Singh, S., 2020. Segmentation based image compression of brain magnetic resonance images using visual saliency. *Biomedical Signal Processing and Control*, Vol. 62, Article 102089, pp. 1–11.
- [25] Taha, A. A. and Hanbury, A., 2015. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, Vol. 15, Article 29, pp. 1–28.
- [26] Taubman, D. S., Marcellin, M. W. and Rabbani, M., 2002. JPEG2000: image compression fundamentals, standards and practice. *Journal of Electronic Imaging*, Vol. 11, Issue 2, pp. 286–287.
- [27] Tong, G., Liu, S., Lv, Y., Pei, H. and Fan, F.-L., 2026. A survey on medical image compression: from traditional to learning-based approaches. *arXiv preprint arXiv:2507.10615 [eess.IV]*, pp. 1–32. Available at: <https://arxiv.org/abs/2507.10615>
- [28] Tunga, P. P. and Singh, V., 2021. Compression of MRI brain images based on automatic extraction of tumor region. *International Journal of Electrical and Computer Engineering (IJECE)*, Vol. 11, Issue 5, pp. 3964–3976.

- [29] Ungureanu, V.-I., Negirla, P. and Korodi, A., 2024. Image-compression techniques: classical and “region-of-interest-based” approaches presented in recent papers. *Sensors*, Vol. 24, Issue 3, Article 791, pp. 1–32.
- [30] Vijaya, S. M., Veena, M. N. and Vinutha, B. A., 2024. Hybridization of lossless and lossy compression for medical images. *International Journal of Computer Applications*, Vol. 184, Issue 32, pp. 1–6.
- [31] Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P., 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, Vol. 13, Issue 4, pp. 600–612, doi: 10.1109/TIP.2003.819861.
- [32] Zou, K., Chen, Z., Yuan, X., Shen, X., Wang, M. and Fu, H., 2023. A review of uncertainty estimation and its application in medical imaging. *Meta-Radiology*, Vol. 1, Issue 1, Article 100003, pp. 1–18, ISSN 2950-1628.

APPENDIX-I

SELECTED SOURCE CODE LISTINGS

This appendix collects the longer code listings referenced from the main chapters. The listings are reproduced exactly as they appear in the working scripts, with only minor formatting changes for readability on the page. The complete source code, together with the trained model weights (roi_unet.h5 and compression_ae_roi_aware.h5), is maintained outside the thesis document.

I.1 U-Net Architecture Definition

The U-Net used for tumour segmentation is built with Keras' functional API. The full network is roughly forty lines of code:

```
def build_unet(input_size=(128, 128, 1)):
    inputs = Input(input_size)

    # Encoder
    c1 = Conv2D(32, 3, activation='relu', padding='same')(inputs)
    p1 = MaxPooling2D(2)(c1)

    c2 = Conv2D(64, 3, activation='relu', padding='same')(p1)
    p2 = MaxPooling2D(2)(c2)

    c3 = Conv2D(128, 3, activation='relu', padding='same')(p2)
    p3 = MaxPooling2D(2)(c3)

    # Bottleneck
    b = Conv2D(256, 3, activation='relu', padding='same')(p3)

    # Decoder with skip connections
    u3 = UpSampling2D(2)(b)
    u3 = concatenate([u3, c3])
    d3 = Conv2D(128, 3, activation='relu', padding='same')(u3)

    u2 = UpSampling2D(2)(d3)
    u2 = concatenate([u2, c2])
    d2 = Conv2D(64, 3, activation='relu', padding='same')(u2)

    u1 = UpSampling2D(2)(d2)
    u1 = concatenate([u1, c1])
    d1 = Conv2D(32, 3, activation='relu', padding='same')(u1)

    outputs = Conv2D(1, 1, activation='sigmoid')(d1)

    return Model(inputs=inputs, outputs=outputs)
```

Listing I.1 U-Net architecture in Keras functional API.

I.2 Combined Dice + Focal Loss

The loss function used to train the U-Net is the sum of a Dice loss term and a focal cross-entropy term. The Dice term drives overlap, and the focal term focuses learning on hard-to-classify pixels.

```
def dice_loss(y_true, y_pred, eps=1e-6):
    y_true = tf.cast(y_true, tf.float32)
    y_pred = tf.cast(y_pred, tf.float32)
    num = 2.0 * tf.reduce_sum(y_true * y_pred) + eps
    den = tf.reduce_sum(y_true) + tf.reduce_sum(y_pred) + eps
    return 1.0 - num / den

def focal_loss(y_true, y_pred, alpha=0.75, gamma=2.0):
    y_true = tf.cast(y_true, tf.float32)
    y_pred = tf.clip_by_value(y_pred, 1e-8, 1.0 - 1e-8)
    fl = - alpha * y_true * tf.pow(1.0 - y_pred, gamma) *
    tf.math.log(y_pred)
    return tf.reduce_mean(fl)

def combined_loss(y_true, y_pred):
    return dice_loss(y_true, y_pred) + focal_loss(y_true, y_pred)
```

Listing I.2 Combined Dice + Focal loss function.

I.3 Autoencoder Definition

The background-compression autoencoder uses stride-2 convolutions to downsample the input three times into a $16 \times 16 \times 4$ latent. The decoder mirrors this with upsampling and convolutions.

```
def build_compression_ae(input_shape=(128, 128, 1)):
    inputs = Input(input_shape)

    # Encoder
    x = Conv2D(32, 4, strides=2, padding='same',
activation='relu')(inputs)
    x = Conv2D(64, 4, strides=2, padding='same', activation='relu')(x)
    latent = Conv2D(4, 3, strides=2, padding='same',
activation='relu', name='latent')(x)

    # Decoder
    x = UpSampling2D(2)(latent)
    x = Conv2D(64, 3, padding='same', activation='relu')(x)
    x = UpSampling2D(2)(x)
    x = Conv2D(32, 3, padding='same', activation='relu')(x)
    x = UpSampling2D(2)(x)
    outputs = Conv2D(1, 3, padding='same', activation='sigmoid')(x)

    return Model(inputs=inputs, outputs=outputs,
name='compression_ae')
```

Listing I.3 Convolutional autoencoder architecture.

I.4 Margin-Sweep Evaluation Loop

The margin sweep iterates over $m \in \{0, 2, 4, 6, 8, 12\}$, runs the full inference pipeline at each margin, and aggregates coverage, leakage, PSNR and SSIM into a results table.

```
MARGINS = [0, 2, 4, 6, 8, 12]
rows = []

for m in MARGINS:
    cov_list, leak_list, psnr_list, ssim_list = [], [], [], []

    for img, gt in zip(X_test, Y_test):
        prob = unet.predict(img[None], verbose=0)[0, ..., 0]
        roi_safe = process_roi(prob, margin=m)

        bg = img * (1 - roi_safe[..., None])
        comp = compress_background(bg, margin=m)
        rec = reconstruct(img, roi_safe, comp)

        cov = np.sum(gt * roi_safe[..., None]) / (np.sum(gt) + 1e-8)
        leak = 1.0 - cov
        psnr = psnr_metric(img, rec, data_range=1.0)
        ssim = ssim_metric(img.squeeze(), rec.squeeze(),
data_range=1.0)

        cov_list.append(cov); leak_list.append(leak)
        psnr_list.append(psnr); ssim_list.append(ssim)

    rows.append({
        'margin': m,
        'coverage': np.mean(cov_list),
        'leakage': np.mean(leak_list),
        'psnr': np.mean(psnr_list),
        'ssim': np.mean(ssim_list),
    })
```

Listing I.4 Margin sweep evaluation loop.

APPENDIX-II

SUPPLEMENTARY RESULTS AND FIGURES

This appendix collects results that did not fit in the main chapters but are useful for completeness. Figure and table numbers are local to the appendix .

II.1 Per-patient Margin Sweep Statistics

The numbers reported in Chapter 4 are averaged over all test slices. For completeness, Table II.1 shows the inter-patient variation in coverage and leakage at margin $m = 6$. The mean is the value reported in Section 4.6; the standard deviation gives a sense of how much variation there is between patients.

Statistic	Coverage at $m = 6$	Leakage at $m = 6$
Mean	0.959	0.041
Std deviation	0.031	0.031
Minimum (worst case)	0.872	0.128
Maximum (best case)	0.998	0.002
Median	0.965	0.035

Table II.1 Coverage and leakage variation across patients at margin $m = 6$.

II.2 Qualitative Reconstruction Examples

Figure II.1 shows four representative test slices reconstructed at margins $m = 0$, 4, 8 and 12. Each row corresponds to one slice; the columns are: original image, predicted ROI mask, reconstruction at $m = 0$, $m = 4$, $m = 8$ and $m = 12$. The tumour region inside the safe mask is preserved exactly in all reconstructions; the differences between columns reflect the increasing background blurring as m grows.

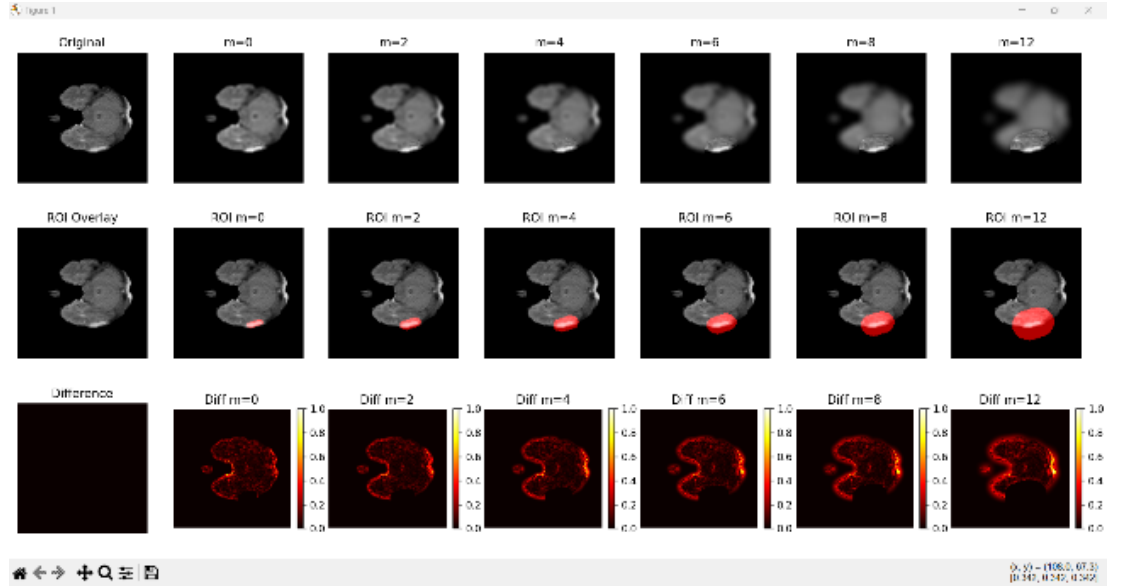


Fig. II.1 Reconstruction examples at four margin values for four representative slices.

II.3 Failure Cases

A small number of slices in the test set produce noticeably worse reconstructions than the averages would suggest. These fall into three categories:

- Slices with very small tumours ($r < 0.01$) where the U-Net produces a partially correct mask but the safety margin pulls in surrounding healthy tissue. The reconstruction is fine but the effective compression ratio drops because the safe mask is larger than the actual tumour by an order of magnitude.
- Slices with multiple disconnected tumour regions, where the segmenter sometimes detects only the largest component. The safety margin around the detected component does not help the missed components. This case is rare in BraTS but is worth flagging for clinical deployment.
- Slices near the top or bottom of the volume where the brain occupies only a small fraction of the slice. The intensity normalisation pushes background noise up and the autoencoder produces visibly noisier reconstructions of these regions.

None of these failure modes is unique to the proposed framework; they are general limitations of slice-by-slice 2D processing and would be partly fixed by moving to a 3D pipeline.

LIST OF PUBLICATIONS AND THEIR PROOFS

The papers communicated from this thesis work are listed below in chronological . Both papers are at the communicated/under-review stage as of the submission of this thesis.

1. Sehrawat, S. and Jayanthi, N., 2026. ROI-aware deep learning framework for brain MRI compression. Communicated to an international peer-reviewed conference (under review).
2. Sehrawat, S. and Jayanthi, N., 2026. Margin-aware ROI expansion for robust brain MRI compression under segmentation uncertainty. Communicated to an international peer-reviewed conference (under review).



DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)
Shahbad Daultapur, Main Bawana Road, Delhi-42

PLAGIARISM VERIFICATION

Title of the Thesis: _____

Total Pages: _____ Name of the Scholar: **Sidharth Sehrawat (24/SPDD/19)**

Supervisor(s):

(1) _____

(2) _____

(3) _____

Department: **Electronics and Communication Engineering**

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: _____ Similarity Index: _____%,

Total Word Count: _____

Date: _____

Candidate's Signature

Signature of Supervisor

CURRICULUM VITAE / BRIEF PROFILE

SIDHARTH SEHRAWAT

M.Tech (Signal Processing and Digital Design) | Roll No. 2K24/SPD/19

Academic Background

- M.Tech, Signal Processing and Digital Design, Delhi Technological University, expected May 2026
- B.Tech, Electronics and Communication Engineering

Areas of Interest

- Medical image processing and compression
- Deep learning for image segmentation
- Signal processing and learned compression

Communicated Publications from Thesis Work

- ROI-aware deep learning framework for brain MRI compression, communicated, 2026.
- Margin-aware ROI expansion for robust brain MRI compression under segmentation uncertainty, communicated, 2026.

Tools and Frameworks

- Languages: Python, MATLAB, C/C++
- Frameworks: TensorFlow / Keras, PyTorch, OpenCV, NumPy, scikit-image, NiBabel
- Tools: Git, LaTeX, Jupyter, Linux command-line workflows