




OPEN

Navigating the nuances: comparative analysis and hyperparameter optimisation of neural architectures on contrast-enhanced MRI for liver and liver tumour segmentation

Felix Quinton¹, Benoit Presles¹, Sarah Leclerc¹, Guillaume Nodari², Olivier Lopez³, Olivier Chevallier³, Julie Pellegrinelli², Jean-Marc Vrigneaud^{1,2}, Romain Popoff^{1,2}, Fabrice Meriaudeau¹ & Jean-Louis Alberini^{1,2}

In medical imaging, accurate segmentation is crucial to improving diagnosis, treatment, or both. However, navigating the multitude of available architectures for automatic segmentation can be overwhelming, making it challenging to determine the appropriate type of architecture and tune the most crucial parameters during dataset optimisation. To address this problem, we examined and refined seven distinct architectures for segmenting the liver, as well as liver tumours, with a restricted training collection of 60 3D contrast-enhanced magnetic resonance images (CE-MRI) from the ATLAS dataset. Included in these architectures are convolutional neural networks (CNNs), transformers, and hybrid CNN/transformer architectures. Bayesian search techniques were used for hyperparameter tuning to hasten convergence to the optimal parameter mixes while also minimising the number of trained models. It was unexpected that hybrid models, which typically exhibit superior performance on larger datasets, would exhibit comparable performance to CNNs. The optimisation of parameters contributed to better segmentations, resulting in an average increase of 1.7% and 5.0% in liver and tumour segmentation Dice coefficients, respectively. In conclusion, the findings of this study indicate that hybrid CNN/transformer architectures may serve as a practical substitute for CNNs even in small datasets. This underscores the significance of hyperparameter optimisation.

Medical image segmentation is a crucial and extensive research domain, acknowledged in both the computer vision and medical image analysis communities¹. It plays a critical role throughout the healthcare process, including clinical diagnosis, treatment planning and follow-up¹⁻³.

Accurate tumour segmentation is crucial in the context of Selective Internal Radiation Therapy (SIRT). SIRT is a specialised treatment approach commonly utilised for liver tumours, whereby radioactive microspheres are delivered directly into the blood vessels supplying the tumour, precisely targeting it while sparing healthy tissue. In this procedure, an exact segmentation of the liver and the tumour ensures an optimal dosimetry calculation, which leads to the efficacy and safety of the SIRT treatment. Therefore, enhancing segmentation results in more accurate dosimetry⁴, leading to a more efficient treatment approach.

In recent years, deep learning methods have emerged as the primary approach to achieve state-of-the-art results in various medical image segmentation operations, including organ and tumour segmentation⁵⁻⁷. This aspect makes them relevant for SIRT treatment planning.

¹Institut de Chimie Moléculaire de l'Université de Bourgogne, ICMUB UMR CNRS 6302, Université Bourgogne, 21000 Dijon, France. ²Service de Médecine Nucléaire, Centre Georges-François Leclerc, 21000 Dijon, France. ³Service de Radiologie et Imagerie Médicale Diagnostique et Thérapeutique, Centre Hospitalier Universitaire, 21000 Dijon, France. ✉email: felix.quinton@u-bourgogne.fr

Initially, architectures based on Convolutional Neural Networks (CNNs) designed for classification, including AlexNet⁸, ResNet⁹, VGG¹⁰, and Inception¹¹, were adapted and used as basic units for segmentation. Subsequently, the structured “U-shaped” encoder-decoder model, exemplified by U-Net¹², was introduced. This method has proven to be an efficient answer to the problems of semantic segmentation, consistently yielding excellent results. However, there has been a recent application of transformer-based structures¹³, originating from natural language processing, to the field of computer vision, resulting in a new group of architectures exhibiting superior performance as compared to CNN architectures in certain tasks^{14,15}. Transformer-based architectures have been applied in medical imaging^{16–18}. Nonetheless, the superiority of transformers compared to CNNs has not been established yet in small datasets, where CNNs can still outperform transformer-based architectures due to the requirement of a large amount of data to exploit the full capacity of this type of architecture. Thus, the circumstances in which transformers can outperform CNNs are unclear. This proliferation of models complicates the selection of a candidate for a specific application.

In addition to architectures, the training strategy can also impact model performance. In this paper, the combinations of the proposed architecture and the proposed training strategy will be referred as a pipeline. Figure 1 illustrates the most common categories and subcategories involved in a tailored deep learning pipeline for medical imaging. Each of these choices can lead to noteworthy improvements in the precision and clinical utility of the final network output.

Now, examining individually each block illustrated in Fig. 1, pre-training a model⁸ enables the model to initialise with weights adjusted to a similar data distribution, potentially leading to faster convergence and improved performance. The sub-field of pre-training known as self-supervised learning¹⁹ is highly beneficial in medical imaging. Unlike pre-training, it can be implemented on images without annotations²⁰.

Handling 3D medical images that potentially exhibit noteworthy differences between images necessitates data preprocessing in order to standardise the dataset. Such preprocessing usually includes procedures such as normalising intensity levels and re-sampling to guarantee that the input data are of homogeneous throughout the entire dataset.²¹

Data augmentation methods^{22–25} including random rotation, scaling, and flipping can be used to artificially enhance the size of the training dataset and enhance the generalisation capabilities of models.

Learning paradigms are utilised to regulate model progression in training. In particular, supervised loss functions play a crucial role in computing the difference between predictions and labels. Subject to the task and intended results, the choice of loss function varies. The most commonly employed options include cross-entropy-based and dice coefficient-based losses^{26,27}. Similarly, the chosen optimiser can greatly affect both the speed of training and the final performance. Its role is to optimise the model weights based on the derivative value of the loss function in order to minimise the value of the function itself. Among various optimisers, stochastic gradient descent (SGD)²⁸, Adam²⁹, and AdamW³⁰ are the most popular choices.

Finally, the segmentation results can be refined and any remaining artefacts or noise can be removed by using post-processing techniques, such as region growing³¹, conditional random fields³², or mathematical morphology operations, including dilation, erosion, opening, or keeping the largest connected component.

When deep learning pipelines are published, they are typically optimised and designed for a specific task, which can lead to a decrease in performance when applied to different tasks or datasets. In an effort to address this issue, frameworks like nnUNet³³ have been introduced to enhance generalisation across heterogeneous datasets. Building on this idea, the objective of this study is to combine and compare the training strategies of seven promising deep learning pipelines for 3D medical image segmentation specifically within the context of SIRT treatment planning, with a focus on liver and tumour segmentation. To this end, this study combines the different elements of the seven pipelines into a single one, and optimises the performance of the seven corresponding architectures by varying and adapting the value of each hyperparameter using Bayesian search. With this approach, the study aims to identify the most effective strategies for 3D liver tumour segmentation. This study focuses on network selection, pre-processing, data augmentation and learning paradigm. All models are trained and optimised on the A Tumor and Liver Automatic Segmentation (ATLAS)³⁴ dataset, which consists of 3D contrast-enhanced magnetic resonance images (CE-MRI) of the liver and tumour with annotations for patients presenting hepatocellular carcinoma (HCC).

Several studies using comparable datasets have been documented in the scientific literature but on private datasets. Christ et al.³⁵ performs Diffusion-Weighted MRI (DW-MRI) segmentation on 31 patients with HCC using cascaded fully convolutional neural networks, Zhao et al.³⁶ proposed liver tumour detection through the using of generative adversarial networks on 131 patient with HCC, and similarly, Kim et al.³⁷ performed

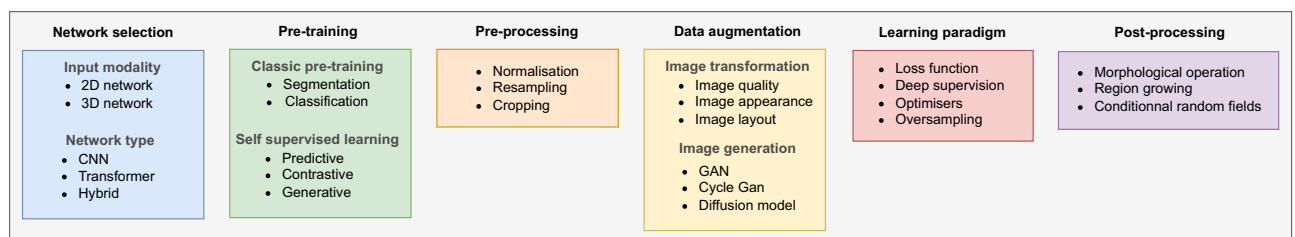


Figure 1. A visual representation of a deep learning pipeline for medical imaging, divided into six key stages: 1) Network Selection, 2) Pre-training, 3) Pre-processing, 4) Data Augmentation, 5) Learning paradigm, and 6) Post-processing.

region-of-interest detection of HCC on a multi-centre CE-MRI dataset containing 545 patients. Xiao et al.³⁸ introduced a liver tumour segmentation solution employing radiomic features from T2 delay-phase CE-MRI over 200 patients. Zhao et al.³⁹ contributed to liver tumour segmentation on multi-modal non-contrast MRI on 255 HCC patients, while Zheng et al.⁴⁰ leveraged multi-phase dynamic 4D CE-MRI for segmentation on a dataset including 190 HCC patients.

This study distinguishes itself from the existing literature by providing a comprehensive and unbiased comparison of the most promising architectures for 3D medical image segmentation. In addition, it explores the impact of different training strategies. In particular, this work goes further than previous publications by examining the impact of each hyperparameter in relation to both the other hyperparameters and the network architecture. This approach provides a more nuanced understanding of the factors that contribute to the performance of deep learning models for medical image segmentation.

The main contribution of this work can be summarised as follows:

- Training strategies optimisation: This work presents a detailed study of the impact of various training strategies on seven advanced architectures for 3D medical image segmentation.
- Comparative analysis using publicly available CE-MRI data: The study offers a unique comparative analysis of liver and tumour segmentation using publicly available CE-MRI data, distinguishing it from prior research.
- In-depth architectural comparison: It provides a comprehensive comparison of CNN, transformer, and hybrid architectures using the ATLAS dataset, contributing to a better understanding of their relative strengths and weaknesses in medical image segmentation.
- Hyperparameter evaluation: The research evaluates critical hyperparameters and contributes to the understanding of their role in optimising segmentation models.
- Resources: A GitHub repository containing code that facilitates the training and tuning of the seven deep learning architectures for 3D segmentation tasks and reproduces the results: gitlab.in2p3.fr/iftim/public-projects/navigating-the-nuances.

These contributions provide valuable insights into the factors that influence the performance of deep learning models in medical image segmentation along with practical tools and recommendations to improve their accuracy and generalisability. The objective of this study is to determine the optimal configuration (architecture and training strategy) on the ATLAS dataset.

Materials and methods

Dataset

The ATLAS³⁴ dataset was used in this study to design and evaluate the complete optimisation process. The ATLAS dataset was chosen for this study on MRI-based liver tumour segmentation due to the lack of research in this area, unlike CT, where the LiTS⁴¹ dataset is popular. This choice aims to address the research gap, especially since MRI is widely used in clinical environment. The study also compares transformer and CNN models on smaller datasets, exploring the potential of transformers in a field dominated by CNNs. This decision reflects a strategy to provide new insights in under-researched areas and test various neural network architectures in clinically relevant, scenarios.

The dataset is made up of 90 contrast-enhanced magnetic resonance images (CE-MRI), collected from 90 patients with hepatocellular carcinoma (HCC). Alongside the CE-MRI, label images of the liver and tumours were provided. The labels were manually delineated by an experienced MRI radiologist using the MIM SurePlan LiverY90 software⁴² from the transversal view CE-MRIs. As shown in Fig. 2, there are three classes in this dataset: background, liver and tumour.

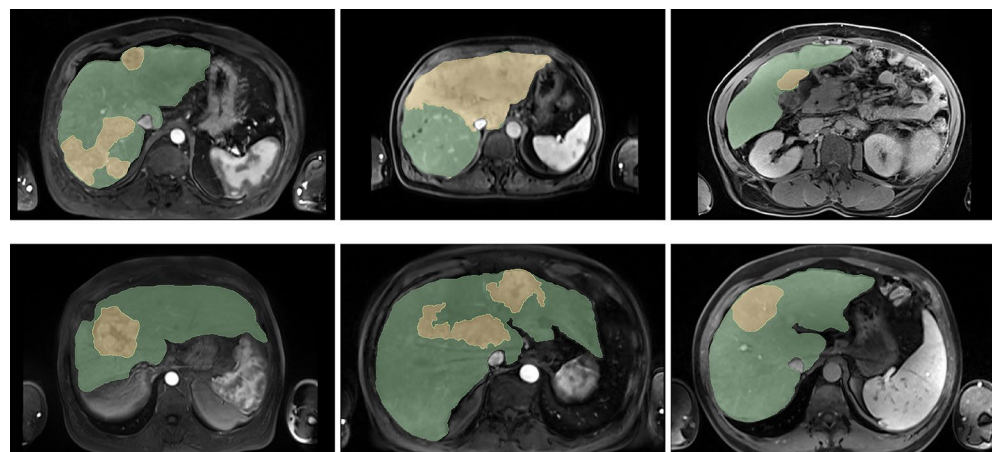


Figure 2. Axial slices of six contrast-enhanced magnetic resonance images from six different patients with the corresponding image labels superimposed. The liver appears in green and tumour in yellow.

Model	Category	Main characteristics
nnFormer ⁴³	Hybrid	Local and global attention blocks, convolutional down-sampling
nnUNet ³³	CNN	U-Net featuring five encoder and decoder blocks
SegmentationNet ⁴⁴	CNN	U-Net featuring four encoder-decoder blocks
Swin-UNet ^{45,46}	Hybrid	Swin-UNet ⁴⁷ based encoder, CNN decoder, pre-trained with SSL
Transbts ⁴⁸	Hybrid	Large transformer bottleneck, CNN encoder/decoder
UNet ⁴⁹	Hybrid	Transformer based encoder, CNN decoder
VT-UNet ⁵⁰	Transformer	Alternates regular and shifted attention windows layer, pre-trained with Swin T ¹⁴

Table 1. Main characteristics of the models studied.

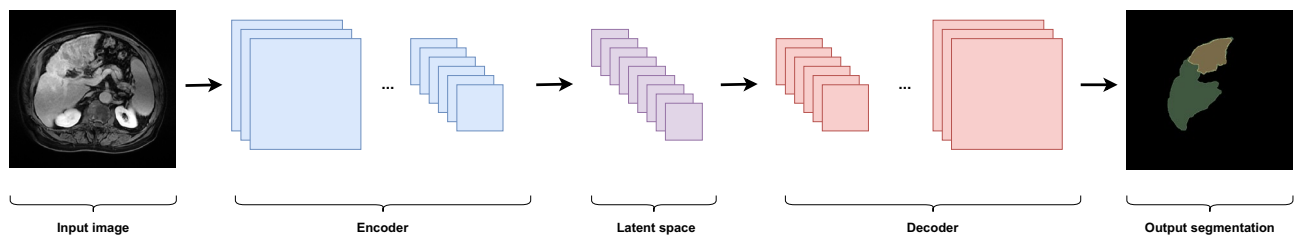


Figure 3. Classic encoder / decoder pattern for image segmentation.

The ATLAS dataset contains 3D CE-MRI of the whole liver and tumour, consisting of 44 to 136 transverse slices of the thorax and abdomen. The pixel spacing of each slice in the dataset ranges from $0.68 \times 0.68 \text{ mm}^2$ to $1.41 \times 1.41 \text{ mm}^2$, with a slice thickness of 2 mm to 4 mm. Bias field correction was applied to every image.

The ATLAS dataset is composed of two distinct sets. The training set which contains 60 images and the testing set with 30 images. To evaluate the performance of each model in this study, we randomly divided the training dataset into two sets: 48 images were used to train the models, and 12 images were reserved to validate them. The test set was used to measure the optimisation process's impact.

Model baselines

This work groups seven different deep learning-based pipelines, optimised for seven different deep learning tasks. Each of the pipelines has demonstrated remarkable efficacy for their respective task.

All the seven chosen architectures have been designed using an encoder/decoder scheme illustrated in Fig. 3, to process 3D images for the purpose of medical image segmentation of MR or CT images. These architectures are described in Table 1. Two are CNNs, one is a pure transformer network, and the remaining four are hybrid CNN/transformer architectures.

To achieve a fair comparison between the seven chosen deep learning architectures, a consistent methodology was introduced. Individual components from every pipeline were extracted and then merged to create a combined pipeline. This procedure enables us to fairly evaluate the impact of each component of the deep learning pipeline on the quality of the segmentation.

Optimisation strategy

To optimise a model's performance through hyperparameters tuning, three optimisation strategies can be considered: random search, grid search, and Bayesian search^{51,52}. Random search involves randomly sampling hyperparameter values within specified ranges, whereas grid search exhaustively explores all possible combinations of hyperparameters. By contrast, Bayesian search is a hyperparameter optimisation technique that utilises a probabilistic model to improve the new hyperparameter combinations based on previous evaluations. The search is guided by information from earlier trials towards promising areas in the hyperparameter space where optimal configurations are more likely to be located. Through iterative selection of new hyperparameter configurations for evaluation based on the model's predictions, Bayesian search effectively reduces the search space and identifies the most suitable hyperparameters for a machine learning model. Thus, all the selected architectures were trained multiple times using various hyperparameter configurations, in compliance with the Bayesian search optimisation strategy from WandB⁵³, which is based on the scikit-learn implementation⁵⁴.

Due to the considerable number of hyperparameter combinations, the optimisation process was carried out in several consecutive phases and some hyperparameters were fixed. This approach extends Bayesian search, providing a further decrease in the number of combinations and enables for a deeper exploration of the impact of similar hyperparameters relative to one another.

Therefore, the studied hyperparameters were categorised into three groups and optimised during three successive optimisation phases: the patch size optimisation phase, the data pre-processing and data augmentation optimisation phase, and the learning paradigms optimisation phase. During each phase, for each of the seven architectures, the same set of hyperparameters was optimised. The optimised configuration of hyperparameter per architecture is then saved and carried forward into the next optimisation phase.

Validation metrics

The model's performance was measured on the validation set at each epoch during training. Due to the incapacity of the models to process large 3D images at once on a 32 GB Graphics Processing Unit (GPU), a sliding window approach was adopted in validation with a 50% overlap. In this study, inspired by the nnUNet pipeline, we utilise an exponential moving average of the Generalised Dice (GD)⁵⁵ score as our validation metric for evaluating segmentation quality. This method is specifically chosen to select models at a phase of their training that consistently exhibit strong generalisation capabilities across the test dataset. The GD metric, adjusted for class imbalance, ensures a balanced assessment between different classes. Using an exponential moving average, we blend both historical and current evaluation of the performances, offering a more comprehensive and stable evaluation of the model's generalisation ability. This approach lessens the risk of over-fitting on a small validation set, thus providing a more trustworthy indicator of the model's performance on unseen data. The GD is defined as follow:

$$GD(A, B) = \frac{2 \sum_{i=1}^N w_i |A_i \cap B_i|}{\sum_{i=1}^N w_i (|A_i| + |B_i|)}, \quad (1)$$

where N is the number of classes (three in our case), A and B are two sets representing the predictions and labels, respectively. A_i and B_i ($\forall i \in \llbracket 1, N \rrbracket$) represent the predictions and labels for class i , respectively and $|\cdot|$ denotes the cardinality or number of elements. The weight w_i ($\forall i \in \llbracket 1, N \rrbracket$) is used to handle class imbalance, and is usually set to the inverse of the square of the number of pixels in each label class $w_i = 1/|B_i|^2$.

With GD_i , the generalised dice at epoch i , the validation metric (VM) at the same epoch is defined as:

$$VM_0 = GD_0 \quad (2)$$

$$VM_i = 0.9 \times VM_{i-1} + 0.1 \times GD_i \quad (3)$$

Evaluation metrics

To quantitatively analyse the quality of the segmentation on the test set, the following metrics were selected:

1. *Dice Coefficient* It measures the similarity between two sets. Given a prediction and a ground truth per pixel for a fixed class the Dice is defined as:

$$\text{Dice} = \frac{2 \times TP}{TP + FP + FN}, \quad (4)$$

where TP, FP and FN respectively corresponds to the number true positives, false positives and false negatives.

2. *5mm Surface Dice* This metric provides a variant of the Dice coefficient which is calculated within a 5mm distance from the surfaces of the structures:

$$SD_{5\text{mm}} = \frac{2 \times TP_5}{TP_5 + FP_5 + FN_5}, \quad (5)$$

where TP_5 , FP_5 and FN_5 correspond to the number of true positives, false positives, and false negatives subsets of pixels located within 5mm of the surfaces from the structure studied.

3. *Precision* Also known as the positive predictive value, it quantifies the accuracy of positive predictions. Precision can be defined:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (6)$$

where TP and FP respectively corresponds to the number true positives, false positives.

4. *Recall* Also known as sensitivity or true positive rate, it quantifies the ability to detect positive instances. Recall is defined as:

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (7)$$

where TP and FN respectively corresponds to the number true positives, false negatives.

5. *Hausdorff Distance (HD)* It represents the greatest of all the distances from a point in one set to the closest point in the other set. Mathematically, for two-point sets A and B :

$$HD(A, B) = \max \left(\max_{a \in A} \min_{b \in B} d(a, b), \max_{b \in B} \min_{a \in A} d(a, b) \right), \quad (8)$$

where $d(a, b)$ is the Euclidean distance between points a and b .

To assess the significance of each metric, a Shapiro-Wilk test was first conducted on the paired differences between the results before and after optimisation. Based on the results of this test, either a paired t-test or a Wilcoxon signed-rank test was then applied to each metric to determine its statistical significance.

Experiments

Baseline

Prior to any optimisation, each architecture was trained as described in the published articles. These baseline results are used as a starting point to measure the impact of the optimisation process.

Patch size influence

If studies indicate that a larger patch size leads to improved performance^{56–59}, this gain may be negligible and would result in a significant increase in training time. Therefore, using intermediate patch sizes could potentially be more efficient. To this end, three patch size configurations (small, intermediate, and large) were tested for each architecture.

The largest patch size compatible with a 32GB GPU and a batch size of two was selected. For each patch size, the depth was fixed at 64, which covers a substantial portion of the images along that axis. This is because the images have a median size of 80 pixels in that dimension. Width and height patch sizes vary in different configurations. The intermediate patch size is one-fourth the size of the largest patch size, and the small patch size is 16 times smaller than the largest configuration. Consequently, the overall patch size ranges from $64 \times 64 \times 64$ to $384 \times 304 \times 64$, depending on the architecture.

Table 2 gives the detailed patch sizes used in this study. The values in italic represent the baselines patch sizes that are the closest to those proposed by the authors of the seven pipelines on similar datasets. Since only three patch size combinations were tested, Bayesian search was not applied during this phase.

Data pre-processing and data augmentation influence

Data pre-processing and data augmentation are among the most important aspects of the learning process in deep learning algorithms, as they provide robustness to the models^{21–25}.

The main objective of data pre-processing is to standardise a dataset²¹. Therefore, a range of pre-processing techniques could be employed such as image resampling, image cropping, and image normalisation. The large part of the studied pipelines uses median resampling since it appears as a fair compromise between image quality and memory consumption. Thus, all images were resampled to the median spacing of the dataset: $1.04 \times 1.04 \times 3.00 \text{ mm}^3$ leading to images after resampling between $320 \times 250 \times 44$ pixels ($320 \times 250 \times 132 \text{ mm}$) to $512 \times 512 \times 136$ pixels ($512 \times 512 \times 408 \text{ mm}$). The ATLAS dataset presents a significant class imbalance among the different structures (background \gg liver \gg tumour). To address this, an oversampling strategy⁶⁰ was also fixed for every training iteration.

During the optimisation process, the only studied hyperparameter that affected the pre-processing was the image normalisation strategy. Pixel intensities vary significantly across different images due to the various machines and sequences used to acquire the ATLAS dataset. Consequently, inter-image normalisation strategies were not taken into consideration. However, two intra-normalisation strategies, namely, min–max normalisation and Z-score normalisation, were analysed.

Min–max normalisation consists of re-scaling the pixel intensities of an image between 0 and 1 and is defined as:

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)}, \quad (9)$$

where x is the original pixel intensity value, x' is the normalised pixel intensity value, and X the ensemble of pixel values in the image.

Z-score normalisation, is another widely-used technique to re-scale the pixel intensities of images. Z-score normalisation involves re-scaling the pixel intensities to have zero mean and unit variance and is defined as:

$$x' = \frac{x - \mu(X)}{\sigma(X)}, \quad (10)$$

where x is the original pixel intensity value, x' is the normalised pixel intensity value, $\mu(X)$ is the mean pixel intensity value in the image, and $\sigma(X)$ is the standard deviation of pixel intensity values in the image.

Hyperparameter	Small	Intermediate	Large
nnFormer patch size	$64 \times 64 \times 64$	<i>$128 \times 128 \times 64$</i>	$224 \times 224 \times 64$
nnUNet patch size	$96 \times 96 \times 64$	$192 \times 160 \times 64$	<i>$384 \times 288 \times 64$</i>
Swin-UNet patch size	$64 \times 64 \times 64$	$128 \times 128 \times 64$	<i>$256 \times 224 \times 64$</i>
Transbts patch size	$64 \times 64 \times 64$	<i>$128 \times 128 \times 64$</i>	$224 \times 224 \times 64$
UNet patch size	$96 \times 96 \times 64$	<i>$192 \times 160 \times 64$</i>	$384 \times 304 \times 64$
UNet3d patch size	$96 \times 96 \times 64$	$192 \times 160 \times 64$	$384 \times 304 \times 64$
VT-UNet patch size	$64 \times 64 \times 64$	<i>$128 \times 128 \times 64$</i>	$256 \times 224 \times 64$

Table 2. Details of the different patch sizes tested in this study. Italic correspond to baseline configurations.

Data augmentation enables the artificial expansion of a training dataset, resulting in more resilient algorithms with superior generalisation capabilities. This is particularly crucial when working with medical image datasets, which usually have limited data. Data augmentation can be categorised into two forms, image transformation augmentation²³ and image generation-based augmentation⁶¹. In this study, unlike⁶², only image transformation strategies were considered.

Image transformation-based augmentation involves applying a set of transformation to the images. Transformations such as image flipping, image rotation, intensity scaling, and intensity shifting are widely accepted in the literature⁶² and are already utilised in most of the seven original pipelines. Therefore, they were systematically applied.

In addition to these standard transformations, six other data augmentation transformations were implemented in the seven selected pipelines. To simplify the optimisation process, we divided this augmentation techniques into three distinct groups: the image fidelity group (IF) with Gaussian noise and Gaussian blur, the scaling and resolution group (SR) with zoom and low image quality simulation and the luminance and contrast group (LC) with contrast and gamma (inverted and non-inverted) alterations.

Each group consists of two closely related augmentation methods, that help to reduce the combinatorial complexity. This leads to three hyperparameters to tune during the optimisation process. The detailed parameterisation for each augmentation method is provided in Table 3.

Thus, the impact of four hyperparameters was considered for this optimisation phase (normalisation, image fidelity, scaling and resolution, luminance and contrast). The normalisation parameter being either min–max or z-score and each data augmentation hyperparameter was either on or off, it leads to 16 possible combinations. Eight of these combinations were tested for each architecture using a Bayesian search optimisation strategy. The best hyperparameter configuration identified during the patch size optimisation phase was used for comparison.

Learning paradigms influence

Learning paradigms control a model's progression during the training. In supervised segmentation learning, a loss function quantifies the difference between the model's prediction and a label. Loss functions for medical image segmentation can be categorised into four categories: distribution-based, region-based, boundary-based (they are not covered in this paper), and compound-based²⁷.

Distribution-based losses: the objective of distribution-based loss functions is to reduce the differences between two distributions, namely the predicted and target distributions. The cross-entropy (CE) loss based on the Kullback-Leibler divergence serves as the foundation for other functions in this category. CE loss can be defined as:

$$\text{CE Loss} = -\frac{1}{I} \sum_{j=1}^J \sum_{i=1}^I y_{ij} \log(\hat{y}_{ij}), \quad (11)$$

where I is the number of voxels, J is the number of classes, y_{ij} is the binary indicator for the class j according to the ground truth and \hat{y}_{ij} the probability of the pixel i to belong to the class j according to the prediction.

Region-based losses: region-based losses minimise discrepancies between predicted segmentation and ground truth by optimising overlap. Functions of this category are derived from the Dice loss function, which involves optimising the Dice Similarity Coefficient. This metric is commonly used to evaluate medical image segmentation tasks. The Dice loss is defined as:

Data augmentation	Probability	Range
Fixed hyperparameters		
Rotations	0.1	0 – 45°
Flipping	0.1	–
Intensity shifting	0.1	0 – 0.1
Intensity scaling	0.1	0 – 0.1
Optimised hyperparameters		
Gaussian noise	0.1	–
Gaussian blur	0.1	–
Zoom	0.1	0.7 – 1.4
Low image quality simulations	0.5	0.5 – 1.0
Contrast	0.1	0.75 – 1.25
Gamma non-inverted images	0.3	0.7 – 1.5
Gamma-inverted images	0.1	0.7 – 1.5

Table 3. Detailed parameterisation per augmentation method. The selected values were based on the values used in the selected pipelines.

$$\text{Dice Loss} = 1 - \frac{2}{J} \sum_{j=1}^J \frac{\sum_{i=1}^I y_{ij} \hat{y}_{ij} + \epsilon}{\sum_{i=1}^I y_{ij} + \sum_{i=1}^I \hat{y}_{ij} + \epsilon}, \quad (12)$$

where J represents the number of classes, I the number of pixels in the images, and \hat{y}_{ij} and y_{ij} represent the predicted probability and the ground truth label for each pixel i in the image to belong to class j . The ϵ term is added for numerical stability and is usually set to a small positive value (e.g., $1e - 5$). If a second version of the Dice loss exists with square terms at the denominator, this version will not be discussed here according to the results of Ma et al.²⁷.

Compound losses: compound losses combine previous losses, including losses from different categories. The DiceCE loss³³ is defined as the addition of the CE loss and the Dice loss. The joint use of Dice loss and cross-entropy loss can improve segmentation performance by taking advantage of each loss function²⁷. Combining both loss functions allows to consider both local accuracy and global consistency in segmentation, maximising segmentation similarity while minimising class probability distribution. The DiceCE loss is defined as follows:

$$\text{DiceCE Loss} = 0.5 \times \text{CE Loss} + 0.5 \times \text{Dice Loss} \quad (13)$$

Although the literature presents numerous loss functions across these three categories, this study will exclusively focus on the two loss functions, Dice and DiceCE, as they are the only two loss functions utilised by the seven pipelines analysed.

Choosing a loss function includes deciding whether to include or not the background class in the loss calculation. Consequently, the effect of the background on performance was also investigated.

While a loss function allows the measurement of the difference between ground truth and prediction, the optimiser controls how the model evolves with respect to that difference. The seven pipelines deploy three distinct optimisers: the Stochastic Gradient Descent (SGD) with Nesterov momentum²⁸, the Adaptive Moment Estimation (Adam)²⁹ and AdamW³⁰.

The SGD optimiser computes the gradients for each parameter and modifies the parameters of the model based on a fraction of the gradient dependent on the value of the learning rate. Utilising a momentum term to consider the preceding gradient enhances convergence and retains momentum in a particular direction. The Nesterov version of the momentum algorithm is an enhancement to the basic momentum update as it computes the gradient after the momentum update, resulting in a more precise direction towards the minimum of the loss function.

Adam combines the concepts of momentum and adaptive learning rates in order to maintain two moving averages for every parameter and adjust the learning rates for each parameter as training progresses. The utilisation of these moving averages aids in stabilising the updates and countering the problems of vanishing or exploding gradients, leading to faster and more stable convergence.

AdamW builds upon Adam by introducing a weight decay regularisation term. This term aids in preventing overfitting by applying a penalty to the magnitude of the weights, steering the model towards simpler, more general representations. AdamW decouples the weight decay from the adaptive learning rate updates, allowing the model to use the benefits of Adam while also employing weight decay for improved generalisation.

The impact of the three optimisers is studied in this phase, and their configurations are detailed in Table 4.

Overall, the three parameters (two losses, inclusion or not of the background class, and three optimisers) lead to 12 possible hyperparameter configurations. During this optimisation phase, eight different combinations per model were tested using Bayesian search. The optimal hyperparameter configuration identified during the data pre-processing and data augmentation optimisation phase was used as point of comparison.

Once the optimisation process completed through the three optimisation phases, the best configuration for each model according to the value of the validation metric is evaluated on the test set and compared to the initial baseline configuration.

Implementation Details

All experiments described in this document were conducted using the PyTorch⁶³ library (version 1.11.0) along with the NVIDIA Compute Unified Device Architecture (CUDA) toolkit⁶⁴ (version 11.3.1). The training and inference of the different architectures were performed using Tesla V100S-PCIE-32GB GPUs. Data processing was performed using the Medical Open Network for Artificial Intelligence (MONAI)⁶⁵ frameworks (version 0.8.1).

Regarding the hyperparameters not mentioned so far, the models were trained in their original configuration optimised by the authors for their own datasets. No post-processing was applied.

Optimiser	Learning rate	Regression weight
Adam	$1e^{-4}$	$3e^{-5}$
AdamW	$1e^{-4}$	$1e^{-2}$
SGD	$1e^{-2}$	$3e^{-5}$

Table 4. Optimiser configuration.

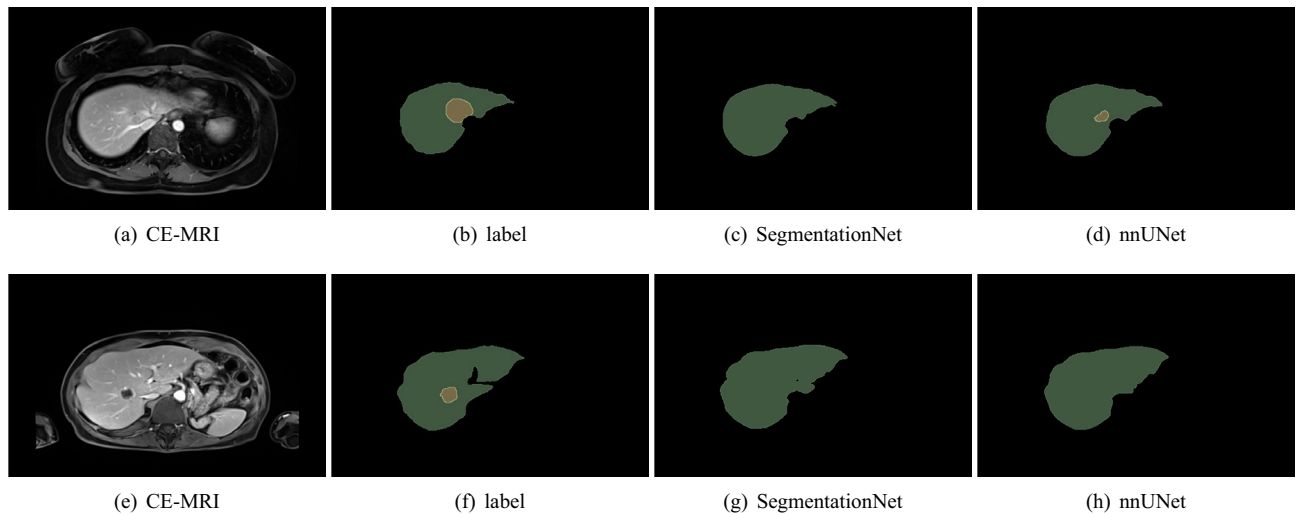


Figure 4. Complex tumour segmentation cases with SegmentationNet and nnUNet. The liver appears in green, and the tumour in yellow.

Model	VMV		
	Small	Intermediate	Large
nnFormer	64.2	<i>64.1</i>	70.4
nnUNet	63.1	63.5	69.0
SegmentationNet	50.3	<i>60.2</i>	63.6
Swin-UNETR	52.3	59.8	60.4
TransBTS	44.4	49.6	56.1
UNETR	36.3	39.4	37.1
VT-UNet	54.8	58.1	56.4
Average	52.2 ± 9.2	56.4 ± 8.2	59.0 ± 10.3

Table 5. Performance per model over validation metric value (VMV) per patch size. Baselines appear in *Italic* and best-performing configurations in **bold**.

Results

Baseline

When examining the baseline inter-model performance outlined in Table 5, a significant discrepancy is observed. There is a considerable gap in performance as the validation metric value ranges from 39.4 points for UNETR to 69.0 points for nnUNet. Whilst UNETR appears to be a performance outlier, there remains a difference of 19.4 points between nnUNet and Swin-UNETR, the second lowest performing model.

Figure 4 highlights that the majority of the models struggle to accurately identify the tumour's location in the most complex cases. In one of the two images, nnUNet is able to classify a few pixels as tumour, but mid-range models such as SegmentationNet are unable to correctly classify any pixels as tumour.

Patch size influence

The effect of the influence of the patch size on the validation set is detailed in Table 5. Online Appendices Tables S1 and S2 provide the test set results. Analysis of intra-model performance reveals that, for five of the seven architectures tested, larger patch sizes correspond to higher performance on the validation set. Regarding the validation metric value, there is an average difference of 2.6 points when comparing the performance of the larger patch size to the intermediate patch size and up to 5.5 points for nnUNet. Nevertheless, for two of the seven architectures (namely UNETR and VT-UNet), a gain in performance is observed with an intermediate patch size. For all architectures except nnFormer, the small patch size results in lower performance than the intermediate and large configurations.

Looking at Fig. 5, it can be observed that smaller patch sizes lead to a significantly higher number of artefacts, even for the best performing architectures. In particular, most segmentation models tend to misclassify the spleen as the liver when using small patch sizes. Larger patches lead to an improvement in the delineation of liver and liver tumour. Nevertheless, even with a large patch size, low-performing models encounter difficulty in eliminating all artefacts.

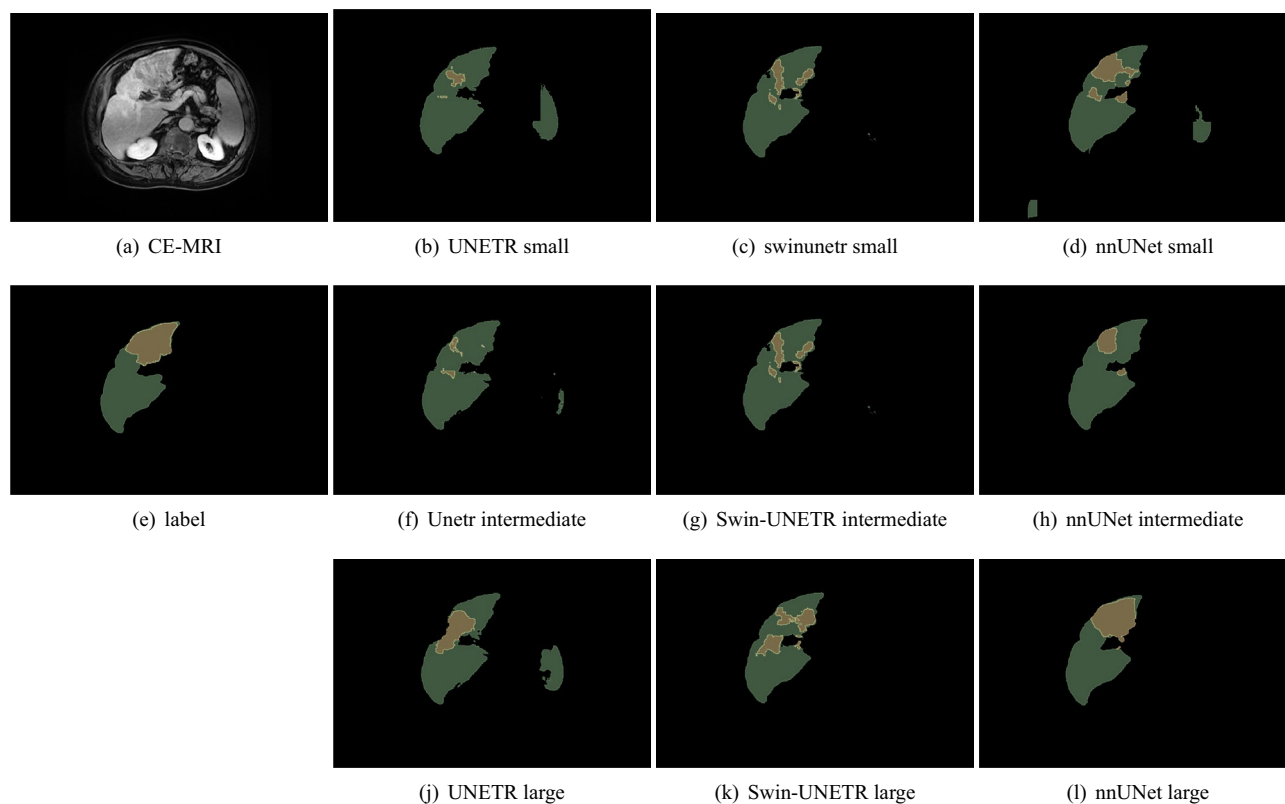


Figure 5. Illustration of the impact of the patch size on the segmentation performance for three different models: UNETR, Swin-UNETR and nnUNet.

Pre-processing and data augmentation influence

The effects of optimising the pre-processing and data augmentation for each model on the validation set are summarised in Table 7 with the best configuration per architecture in Table 6. The detailed results on the test set are given in Online Appendices Tables S3 and S4.

In terms of validation metric value, this optimisation leads to an average gain of 2.5 points when compared to the previous phase. Specifically, TransBTS and UNETR models experienced gains above four points. Nevertheless, a decrease in performance of 1.3 points can be observed for the nnFormer architecture.

Regarding the impact of each parameter across all tested combinations on the validation metric value, min–max normalisation results in an average gain of 2.3 points as compared to the standard-score normalisation. Incorporating scaling and resolution augmentations led to an average gain of 1.7 points and the image fidelity augmentation in an average gain of 1.9. However, the implementation of luminance and contrast augmentations caused a loss of 2.9 points. Figure 6 shows the variation of the performance of each tested combination per architecture during this phase.

Learning paradigm influence

The effects of optimising the learning paradigm for each model are summarised in Table 9 and the best configuration in Table 8. Detailed results on the test set can be found in Online Appendices Tables S5 and S6.

Model	Norm	IF	SR	LC
nnFormer	Range/std	✓	✓	✓
nnUNet	Min/max	✗	✗	✓
SegmentationNet	Min/max	✓	✗	✗
Swin-UNetr	Min/max	✗	✓	✓
Transbts	Min/max	✗	✓	✓
UNetr	Range/std	✗	✓	✓
VT-UNet	Range/std	✗	✗	✗

Table 6. Selected data pre-processing and data augmentation configuration per architecture with normalisation (norm), image fidelity (IF), scaling and resolution (SC) and luminance and contrast (LC) groups.

Model	Initial VMV	New VMV	Difference
nnFormer	70.4	69.1	-1.3
nnUNet	69.0	69.7	+0.7
SegmentationNet	63.6	66.3	+3.3
Swin-UNETR	60.4	63.2	+2.8
TransBTS	56.1	60.4	+4.3
UNETR	39.4	43.8	+4.4
VT-UNet	58.1	61.1	+3.0
Average	59.6 ± 9.6	61.9 ± 8.1	+2.5

Table 7. New validation metric value (VMV) obtained during the pre-processing and data augmentation optimisation compared to the initial VMV obtained after the patch size optimisation.

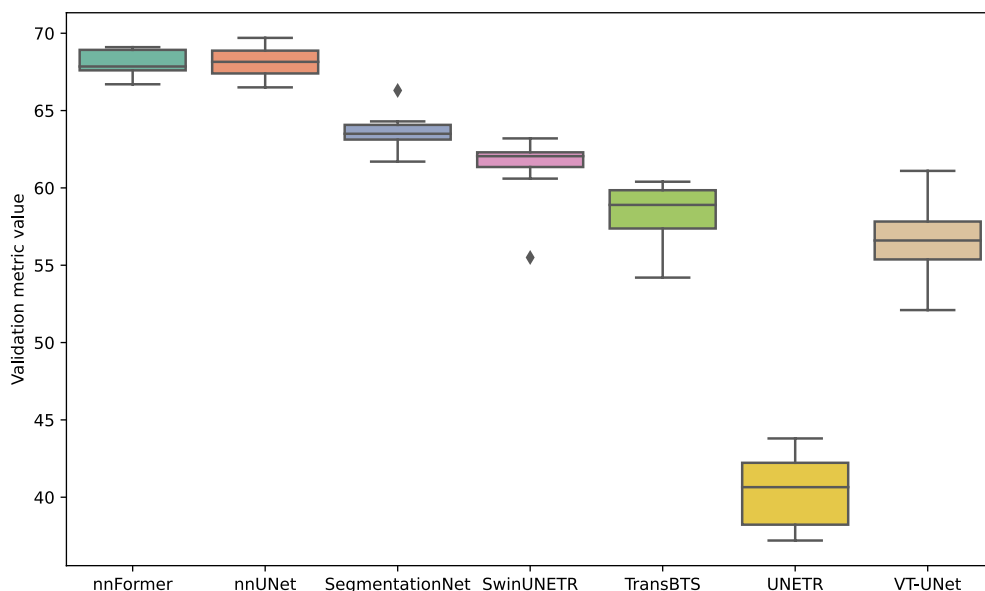


Figure 6. Performance variability across the validation metric value for seven deep learning architectures during the pre-processing and data augmentation optimisation phase.

Model	Loss	Optimiser	Include background
nnFormer	DiceCE	SGD	✗
nnUNet	DiceCE	Adam	✗
SegmentationNet	DiceCE	Adam	✓
Swin-UNetr	DiceCE	AdamW	✓
Transbts	Dice	SGD	✗
UNetr	Dice	Adam	✗
VT-UNet	DiceCE	AdamW	✗

Table 8. Selected learning paradigm configuration per architecture.

In terms of the validation metric value, this optimisation results in an average gain of 0.6 points in comparison to the previous phase. However, for four out of the seven architectures, there is a reduction in performance up to 2.6 points for VT-UNet. Nevertheless, this phase results in a gain of 6.0 points for TransBTS.

Looking at the average of all the parameter combinations studied during this phase, the use of the Adam optimiser led to an average improvement of 1.4 points over AdamW and 2.6 points over SGD based on the validation metric value. Notably, the DiceCE loss outperformed the Dice loss with an average gain of 0.9 points. On the other hand, the choice to incorporate the background in the loss calculation appears to have an insignificant

Model	Initial VMV	Final VMV	Difference
nnFormer	70.4	69.7	-0.7
nnUNet	69.7	70.9	+1.2
SegmentationNet	66.3	65.2	-1.1
Swin-UNETR	63.2	62.8	-0.4
TransBTS	60.4	66.4	+6.0
UNETR	43.8	45.8	+2.0
VT-UNet	61.1	58.5	-2.6
Average	62.1 ± 8.3	62.8 ± 7.9	+0.6

Table 9. Final validation metric value (VMV) obtained after the learning paradigm optimisation phase compared to the initial VMV obtained after the pre-processing and data augmentation phase.

impact on the models' performance, with an average difference of only 0.2 points supporting its inclusion. Figure 7 shows the variation of the performance of each tested combination per architecture during this phase.

Overall progression on the test set

Upon optimising the models, we identified the best hyperparameter sets for each. We then evaluated their performance on the test set. The following sections detail how these models fared in liver and tumour segmentation tasks, offering an analysis of the chosen parameters' efficacy.

In Table 10, we present the progression of the performance throughout the entire optimisation process when testing on the liver dataset. The table highlights the differences between the initial baseline results and the highest performance achieved during training, based on the validation criteria. Improvements can be observed across all models leading to significant improvement for most of the calculated metrics, with an average increase of 1.7% in the Dice coefficient and a reduction of 36.7 mm in the Hausdorff distance. Nevertheless, the improvements in the nnUNet model - the leading baseline for this task - are marginal. Despite varying baseline performances, the optimisation process levelled the playing field, with all seven models achieving Dice coefficients within a narrow range of 92.3–95.1%. Although initially different, the baselines' performances became similar after the optimisation process, as indicated by all seven models presenting a Dice coefficient between 92.3% and 95.1%.

Figure 8 offers a visual representation of the evolution of the performances on the test set between baseline (in red) and optimised (in blue) models on the liver segmentation task. This figure clearly highlights the positive impact of the optimisation process in particular for initially low-performing models. For TransBTS and Swin-UNETR for instance, before optimisation, respectively 12 and 14 images have a dice score below 90% against only one after optimisation.

Table 11 details the progression of performance achieved through the optimisation process on the tumour segmentation task of the test set. The optimisation process enhanced every model's performance, demonstrated

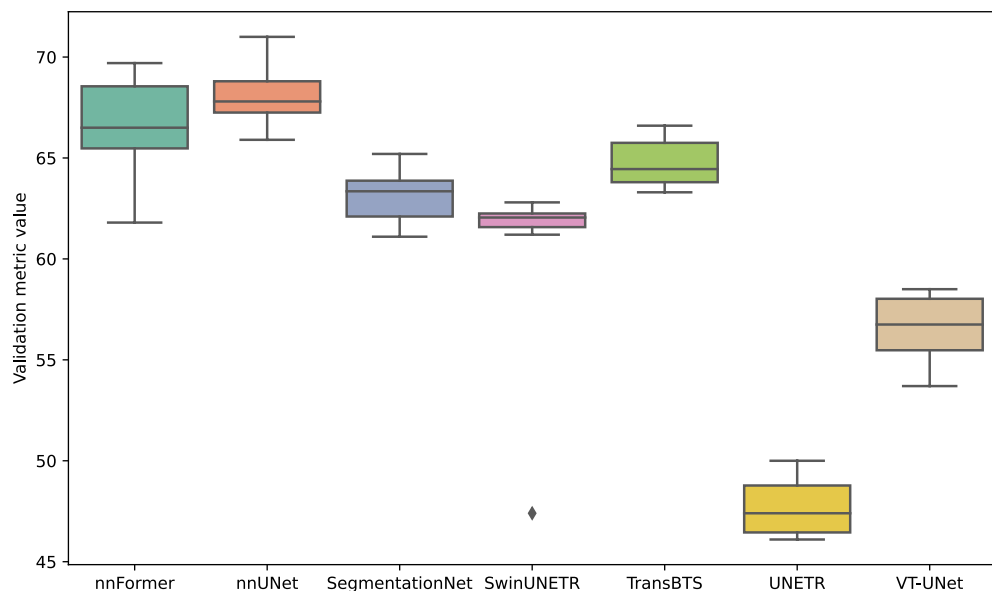


Figure 7. Performance variability across the validation metric value for seven deep learning architectures during the learning paradigm optimisation phase.

Model	Dice			5 mm SD			Precision			Recall			HD (mm)		
	BS	OP	p	BS	OP	p	BS	OP	p	BS	OP	p	BS	OP	p
nnFormer	94.2	95.0	**	91.9	94.5	***	94.6	95.6	*	93.9	94.5	—	72.9	48.8	***
nnUNet	95.1	95.0	—	94.6	95.0	—	94.7	94.9	—	95.4	95.1	—	38.8	39.0	—
SegmentationNet	94.6	94.3	—	93.9	93.7	—	94.8	95.0	—	94.5	93.7	**	52.0	30.0	*
Swin-UNETR	89.6	94.4	***	82.9	93.1	***	85.0	96.4	***	95.4	92.5	***	131.5	42.5	***
TransBTS	90.5	95.1	***	86.7	95.1	***	88.9	95.9	***	92.8	94.4	***	134.0	24.9	***
UNETR	91.9	92.3	—	86.8	87.7	—	91.3	93.1	***	92.9	91.9	***	104.8	117.3	—
VT-UNet	92.8	94.4	**	89.1	93.3	***	91.4	94.6	***	94.6	94.3	—	86.1	41.0	—

Table 10. Performance evolution per model on the test dataset, for the liver between the baseline (BS) and the optimised hyperparameter combination (OP). Italic indicates the best inter-model value for each metric. Stars indicate the level of significance of differences between baseline and optimised results based on paired t-test and Wilcoxon signed-rank test depending on the distribution of the results on the test set according to the Shapiro-Wilk test (no star means not significant, * means $p < 0.05$, ** means $p < 0.01$, and *** means $p < 0.001$).

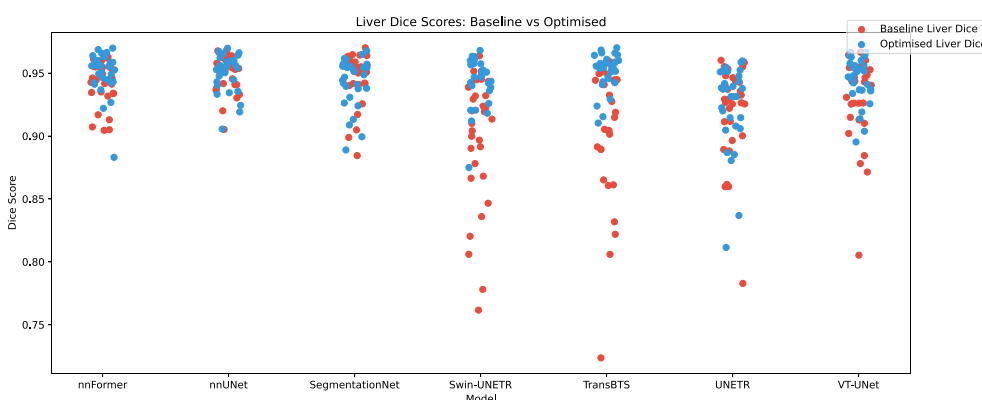


Figure 8. Comparison of the dice performance on liver segmentation of the models across the 30 images of the test set for the seven deep learning architectures between the baseline models and the optimised models.

Model	Dice			5mm SD			Precision			Recall			HD (mm)		
	BS	OP	p	BS	OP	p	BS	OP	p	BS	OP	p	BS	OP	p
nnFormer	68.5	69.7	—	67.6	67.7	—	78.9	82.9	—	65.7	64.4	—	66.7	72.3	—
nnUNet	68.1	68.1	—	67.5	67.8	—	78.8	84.2	—	67.9	63.9	—	60.2	60.1	—
SegmentationNet	55.8	61.7	—	52.5	58.3	—	61.6	74.2	—	55.1	59.2	—	91.5	84.8	—
Swin-UNETR	47.7	55.1	—	42.0	50.9	***	48.7	78.2	*	57.2	48.0	***	162.9	58.5	**
TransBTS	51.4	62.1	**	46.2	59.2	—	58.4	73.1	***	53.6	59.9	*	176.7	73.7	***
UNETR	33.1	41.5	***	30.0	35.3	*	40.9	50.7	—	31.9	39.2	**	121.7	138.1	*
VT-UNet	54.3	56.6	—	49.6	53.9	—	61.1	71.6	—	55.8	56.0	**	98.0	72.7	—

Table 11. Performance evolution on the test dataset, for the tumour between the baseline (BS) and the optimised hyperparameter combination. Italic indicates the best inter-model value for each metric. Stars indicate the level of significance of differences between baseline and optimised results based on paired t-test and Wilcoxon signed-rank test depending on the distribution of the results on the test set according to the Shapiro-Wilk test (no star means not significant, * means $p < 0.05$, ** means $p < 0.01$, and *** means $p < 0.001$).

through higher Dice coefficient and Hausdorff distance scores. On average, the models gained an increase of 4.8% and 31mm in these metrics after the optimisation process. On low-performing models, this is traduced by a significant improvement in performance according to the paired t-test / Wilcoxon signed-rank test on the majority of the metrics. High-performing models such as nnUNet and nnFormer still show a gain in performance. They recorded Dice coefficients of 68.1% and 69.7%, respectively, making them the clear front runners by a considerable margin.

Figure 9 offers a visual representation of the evolution of the performances on the test set between the baseline (in red) and the optimised models (in blue) on the tumour segmentation task. In contrast to the liver

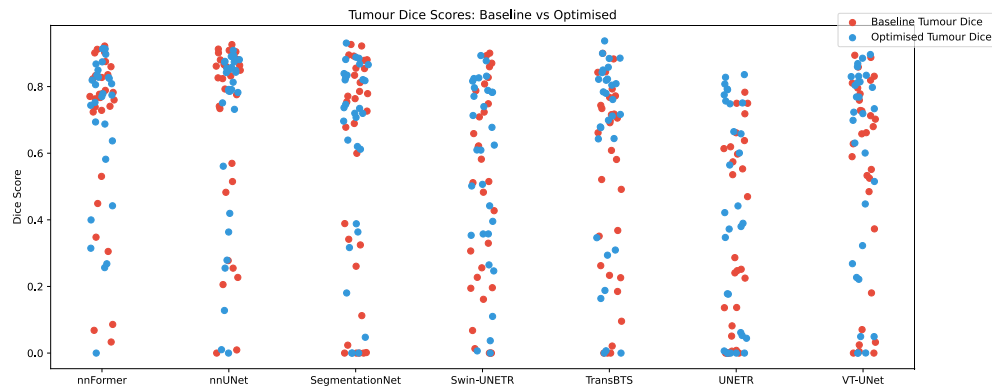


Figure 9. Comparison of the dice performance on tumour segmentation of the models across the 30 images of the test set for the seven deep learning architectures between the baseline models and the optimised models.

segmentation task, the performance per image presents a high variability from one to another ranging from 0% to 95% of Dice.

Discussion

Impact of the different hyperparameters

In this study, a set of hyperparameters has been optimised through Bayesian search for seven distinct deep learning segmentation architectures. The hyperparameters were categorised into three sections: patch size, pre-processing and data augmentation, as well as learning paradigms.

Patch size influence

It is worth noting that traditional CNNs typically allows larger image inputs compared to transformer-based architectures. Thus, with CNN, we were able to load a large part of the images which provide significant context to the network. In contrast, transformers face difficulty to load images with patch sizes larger than $224 \times 224 \times 64$ pixels. Tests on the influence of patch size have revealed that a larger patch size results in a considerable enhancement in overall performance. In medical imaging, a significant context is vital in achieving optimal results. As a consequence, there is a need for more substantial GPUs. However, depending on the training architecture, training models with larger patch sizes for convergence has taken up to twice as much time compared to models with intermediate patch sizes.

Pre-processing and Data Augmentation influence

As illustrated in Fig. 6, the effects of pre-processing and data augmentation appear to rely heavily on the inherent performance capabilities of the particular deep learning model. The resilience of high-performing models like nnFormer and nnUNet is remarkable, as evidenced by their validation metric's relatively low standard deviations. Architectures with lower inherent performance, such as UNETR and VT-UNet, show a significant sensitivity to these strategies. Consequently, customised optimisation methods that are tailored for specific deep learning architectures and contingent on their baseline performance become crucial.

Learning Paradigm influence

In contrast to data augmentation, the influence of the learning paradigm on the baseline performance of the model is not striking, as demonstrated in 7. When examining nnFormer and nnUNet, both models showed a greater sensitivity to these parameters. In contrast, models with lower baseline scores, namely SegmentationNet and TransBTS, exhibited comparatively stable results, potentially possibly indicating less influence of the learning paradigm or saturation of their performance capabilities within the current parameters.

The nuances observed highlight the complexity involved in optimising deep learning architectures. Selecting an appropriate learning paradigm is not simply a matter of best fitting the available data, but requires consideration of the unique characteristics and inherent capabilities of each architecture. A single strategy may not be sufficient, adapting it to the architecture can, in some cases, lead to notable performance improvements.

Overall optimisation impact

Although almost all the model present progress over the different metrics on the liver and tumour segmentation according to Tables 10 and 11, the analysis of Figs. 8 and 9 exposes a different pattern in the impact of the optimisation on the two tasks.

In the context of liver segmentation, the optimisation process not only minimises the number of outliers in lower-performing models but also offers a noticeable improvement in performance, even for images that were already well segmented. Regarding tumour segmentation, most models show a modest improvement for each image analysed. However, it is important to note that images that initially demonstrated low performance prior to optimisation rarely show a significant increase in performance after optimisation. Therefore, these cases can still be classified as outliers.

On an inter-model point of view, after optimisation, if the performance on the liver is similar between the tested models, the performance on tumour is highly variable. Looking at Fig. 9, the images of the test set can be separated into two distinct categories based on performance: those with dice exceeding 60% and those falling below this threshold. While the models tend to exhibit comparable performances in the former group, it is in the latter category where performance disparities become more pronounced. This variance can be attributed to the ability of the best-performing models to excel in handling more complex cases, thus outperforming their average counterparts.

CNN, transformer and hybrid models comparison

This study conducts a categorical comparison between CNNs, transformers, and hybrid models, revealing some noteworthy insights. It is evident that, following optimisation, hybrid networks not only compete with CNNs but can also surpass them, despite having a significantly smaller patch size. Additionally, advancements in GPU technology, particularly in terms of enhanced memory capacity, allowing for larger patch size could further cement hybrid networks as the predominant model in the field. Since VT-UNet stands as the sole purely transformer-based network in this study, making definitive conclusions is challenging. However, the current data suggests that convolution-free models may not yet be a completely reliable alternative.

Literature comparison

In the field of liver and liver tumour segmentation, there is a dearth of studies providing results on mono-modal MRI. Comparisons with other datasets are not consistently applicable due to distinctive imaging protocols and patient groups. Nevertheless, the work by Christ et al.³⁵, as the only study conducted within a similar imaging modality (diffusion weighted MRI), remains relevant. They reported a Dice coefficient of 87% for liver and 69.7% for tumour segmentation across 31 patients. On the liver, the CHAOS segmentation challenge⁶⁶ saw a dice score of 95.2%.

Bayesian search time requirements

The specifics regarding the number of parameters, training duration, and inference time for the optimised configurations are meticulously outlined in Table 12. Each model exhibits an acceptable inference time per image from a clinical perspective, ranging between 0.6 and 3.0 s. However, there is considerable variability in training times, spanning from 17 to 144 h. This wide range could present challenges in the context of Bayesian optimisation.

Although Bayesian search offers an elegant strategy for fine-tuning hyperparameters, it has inherent limitations when it comes to training models in parallel. Unlike grid or random search methods, which can train numerous hyperparameter combinations simultaneously, the Bayesian approach works sequentially while determining the next evaluation based on the findings of previous ones. In our research context, evaluation necessitated testing 19 unique combinations. Due to the maximum training duration of 6 days per combination, the overall process lasted approximately 16 weeks. This restriction poses obstacles especially when dealing with complicated models or extensive datasets, and can markedly extend the tuning phase.

Limitations

The primary objective of this study was to identify the best hyperparameter configuration without utilising external data. However, due to the need to limit the computational expenses, several hyperparameters were intentionally not examined in this study. Here are the limitations of our study:

- **Image generation techniques:** hyperparameters that were not examined in this study involve generating images for data augmentation and the effect of pre-training or self-supervised learning. Although a self-supervised version of SWIN-UNETR was applied, its impact on performance was not assessed. According to Shin et al.⁶⁷, if image generation is achievable without external data, its impact is negligible when other sources of data augmentation are employed
- **Mono-centre dataset:** all images in the ATLAS dataset were obtained from patients at the University Hospital of Dijon, France. As a result, conclusions reached in this study may vary with a more diverse dataset.

Model	nb parameters	Training time (h)	Inference time per image (s)
nnFormer ⁴³	3.7×10^7	144	1.7
nnUNet ⁴³	3.0×10^7	121	0.8
SegmentationNet ⁴⁴	1.8×10^7	28	0.6
Swin-UNetr ^{45,46}	6.2×10^7	77	2.4
Transbts ⁴⁸	3.4×10^7	69	1.1
UNetr ⁴⁹	9.3×10^7	17	0.7
VT-UNet ⁵⁰	1.2×10^7	77	3.0

Table 12. Number of parameters, training time and inference time per model after optimisation.

- **Cross-validation omission:** cross-validation was not utilised in our study due to the extended training period of the deep learning architectures examined. This choice could influence the reliability of our performance metrics and their applicability to diverse data samples.
- **Loss functions:** our focus was limited to the techniques employed in the seven tested pipelines. In particular, we did not examine loss functions customised for specific assignments such as tumour segmentation. These encompassed the Dice topK loss, the boundary loss, and the Hausdorff distance loss.
- **Architecture modifications:** our study refrained from altering the original architectures in relation to depth, width or layer order. The majority of these architectures were developed with datasets of different complexities in mind. Consequently, deviations to these architectures could result in diverse outcomes, particularly when applied to datasets of simpler or more complex natures than originally anticipated.

Conclusion

Our study emphasises the significance of hyperparameter optimisation in the field of medical imaging, particularly in the 3D segmentation of the liver and liver tumour on CE-MRI. By implementing Bayesian search on a subset of hyperparameters, it was possible to clearly measure the effects of patch size, pre-processing, data augmentation and learning paradigms, whilst still keeping the quantity of tested combinations reasonable. An average gain per architecture of 1.7% for the liver and a remarkable 5.0% for the liver tumour using the Dice coefficient illustrates the significance of hyperparameter tuning. Indeed, on such complex tasks, only a few methodological suggestions have resulted in analogous progressions in segmentation quality. However, although among the tested architectures all of them appear to be effective in the literature, it is important to note that significant performance discrepancies have been observed between them, which does not depend on the hyperparameter configurations. Nevertheless, hybrid transformer architectures, typically associated with large datasets, have shown their capability to match CNN performance even with limited data. Thus, for future directions, it appears promising to expand the dataset size through image generation, self-supervised and mixed supervised learning. This is especially true given the potential of transformers to benefit from such expansions. This study encourages a re-evaluation of the role of transformers in scenarios with limited data and highlights their emerging relevance in medical image segmentation. Such an improvement in automatic segmentation represents a significant step towards the potential automation of this process, reducing the reliance on manual segmentation performed by radiologists. This advancement not only streamlines the workflow but also promises to improve patient care by ensuring more precise and timely diagnoses and SIRT treatment planning.

Data availability

The data that support the findings of this study are publicly available at <https://atlas-challenge.u-bourgogne.fr>.

Received: 6 October 2023; Accepted: 1 February 2024

Published online: 12 February 2024

References

1. Lynch, C. J. & Liston, C. New machine-learning technologies for computer-aided diagnosis. *Nat. Med.* **24**, 1304–1305 (2018).
2. Samarasinghe, G. *et al.* Deep learning for segmentation in radiation therapy planning: A review. *J. Med. Imaging Radiat. Oncol.* **65**, 578–595 (2021).
3. Wang, C., Zhu, X., Hong, J. C. & Zheng, D. Artificial intelligence in radiotherapy treatment planning: Present and future. *Technol. Cancer Res. Treat.* **18**, 1533033819873922 (2019).
4. Smits, M. L. *et al.* Radioembolization dosimetry: The road ahead. *Cardiovasc. Intervent. Radiol.* **38**, 261–269 (2015).
5. Liu, X., Song, L., Liu, S. & Zhang, Y. A review of deep-learning-based medical image segmentation methods. *Sustainability* **13**, 1224 (2021).
6. Du, G., Cao, X., Liang, J., Chen, X. & Zhan, Y. Medical image segmentation based on u-net: A review. *J. Imaging Sci. Technol.* **64** (2020).
7. Asgari Taghanaki, S., Abhishek, K., Cohen, J. P., Cohen-Adad, J. & Hamarneh, G. Deep semantic segmentation of natural and medical images: A review. *Artif. Intell. Rev.* **54**, 137–178 (2021).
8. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90 (2017).
9. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 770–778 (2016).
10. Simonyan, K. & Zisserman, A. *Very Deep Convolutional Networks for Large-scale Image Recognition*. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014).
11. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition* 2818–2826 (2016).
12. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III* 18 234–241 (2015).
13. Vaswani, A. *et al.* Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017).
14. Dosovitskiy, A. *et al.* An Image is worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929) (2020).
15. Liu, Z. *et al.* Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* 10012 (2021).
16. Shamshad, F. *et al.* Transformers in medical imaging: A survey. *Med. Image Anal.* 102802 (2023).
17. He, K. *et al.* Transformers in medical image analysis. *Intell. Med.* **3**, 59–78 (2023).
18. Xiao, H., Li, L., Liu, Q., Zhu, X. & Zhang, Q. Transformers in medical image segmentation: A review. *Biomed. Signal Process. Control* **84**, 104791 (2023).
19. Shurrab, S. & Duwairi, R. Self-supervised learning methods and applications in medical imaging analysis: A survey. *PeerJ Comput. Sci.* **8**, e1045 (2022).

20. Tajbakhsh, N. *et al.* Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Med. Image Anal.* **63**, 101693 (2020).
21. De Raad, K. *et al.* The effect of preprocessing on convolutional neural networks for medical image segmentation. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)* 655–658 (2021).
22. Shorten, C. & Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *J. Big Data* **6**, 1–48 (2019).
23. Taylor, L. & Nitschke, G. Improving deep learning with generic data augmentation. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)* 1542–1547 (2018).
24. Hataya, R., Zdenek, J., Yoshizoe, K. & Nakayama, H. Meta approach to data augmentation optimization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* 2574–2583 (2022).
25. Xu, M., Yoon, S., Fuentes, A. & Park, D. S. A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recogn.* 109347 (2023).
26. Milletari, F., Navab, N. & Ahmadi, S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)* 565–571 (2016).
27. Ma, J. *et al.* Loss odyssey in medical image segmentation. *Med. Image Anal.* **71**, 102035 (2021).
28. Ruder, S. *An Overview of Gradient Descent Optimization Algorithms*. arXiv preprint [arXiv:1609.04747](https://arxiv.org/abs/1609.04747) (2016).
29. Kingma, D. P. & Ba, J. *Adam: A Method for Stochastic Optimization*. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014).
30. Loshchilov, I. & Hutter, F. *Decoupled Weight Decay Regularization*. arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101) (2017).
31. Szénási, S., Vámosy, Z. & Kozlovsky, M. Evaluation and comparison of cell nuclei detection algorithms. In *2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES)* 469–475 (2012).
32. Roth, H. R. *et al.* Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part I* 18 556–564 (2015).
33. Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J. & Maier-Hein, K. H. nnu-net: A self-configuring method for deep learning-based biomedical image segmentation. *Nat. Methods* **18**, 203–211 (2021).
34. Quinton, F. *et al.* A tumour and liver automatic segmentation (atlas) dataset on contrast-enhanced magnetic resonance imaging for hepatocellular carcinoma. *Data* **5**, 89 (2023).
35. Christ, P. F. *et al.* *Automatic Liver and Tumor Segmentation of ct and Mri Volumes Using Cascaded Fully Convolutional Neural Networks*. arXiv preprint [arXiv:1702.05970](https://arxiv.org/abs/1702.05970) (2017).
36. Zhao, J. *et al.* Tripartite-gan: Synthesizing liver contrast-enhanced mri to improve tumor detection. *Med. Image Anal.* **63**, 101667 (2020).
37. Kim, J., Min, J. H., Kim, S. K., Shin, S.-Y. & Lee, M. W. Detection of hepatocellular carcinoma in contrast-enhanced magnetic resonance imaging using deep learning classifier: a multi-center retrospective study. *Sci. Rep.* **10**, 9458 (2020).
38. Xiao, X. *et al.* Radiomics-guided gan for segmentation of liver tumor without contrast agents. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part II* 22 237–245 (2019).
39. Zhao, J. *et al.* United adversarial learning for liver tumor segmentation and detection of multi-modality non-contrast mri. *Med. Image Anal.* **73**, 102154 (2021).
40. Zheng, R. *et al.* Automatic liver tumor segmentation on dynamic contrast enhanced mri using 4D information: Deep learning model based on 3d convolution and convolutional lstm. *IEEE Trans. Med. Imaging* **41**, 2965–2976 (2022).
41. Bilic, P. *et al.* The liver tumor segmentation benchmark (lits). *Med. Image Anal.* **84**, 102680 (2023).
42. Mim software. <https://www.mimsoftware.com/>. Accessed: 06 Oct 2023.
43. Zhou, H.-Y. *et al.* *nnformer: Interleaved Transformer for Volumetric Segmentation*. arXiv preprint [arXiv:2109.03201](https://arxiv.org/abs/2109.03201) (2021).
44. Janssens, R., Zeng, G. & Zheng, G. Fully automatic segmentation of lumbar vertebrae from ct images using cascaded 3d fully convolutional networks. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)* 893–897 (2018).
45. Hatamizadeh, A. *et al.* Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 7th International Workshop, BrainLes 2021, Held in Conjunction with MICCAI 2021, Virtual Event, September 27, 2021, Revised Selected Papers, Part I* 272–284 (2022).
46. Tang, Y. *et al.* Self-supervised pre-training of swin transformers for 3d medical image analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 20730–20740 (2022).
47. Cao, H. *et al.* Swin-unet: Unet-like pure transformer for medical image segmentation. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III* 205–218 (2023).
48. Wang, W. *et al.* Transbts: Multimodal brain tumor segmentation using transformer. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I* 24 109–119 (2021).
49. Hatamizadeh, A. *et al.* Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* 574–584 (2022).
50. Peiris, H., Hayat, M., Chen, Z., Egan, G. & Harandi, M. A robust volumetric transformer for accurate 3d tumor segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part V* 162–172 (2022).
51. Marinov, D. & Karapetyan, D. Hyperparameter optimisation with early termination of poor performers. In *2019 11th Computer Science and Electronic Engineering (CEECE)* 160–163 (2019).
52. Bergstra, J., Bardenet, R., Bengio, Y. & Kégl, B. Algorithms for hyper-parameter optimization. *Adv. Neural Inf. Process. Syst.* **24** (2011).
53. Weight and biases. <https://wandb.ai/site>. Accessed: 06 Oct 2023.
54. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
55. Crum, W. R., Camara, O. & Hill, D. L. Generalized overlap measures for evaluation and validation in medical image analysis. *IEEE Trans. Med. Imaging* **25**, 1451–1461 (2006).
56. Pinheiro, P. & Collobert, R. Recurrent convolutional neural networks for scene labeling. In *International Conference on Machine Learning* 82–90 (2014).
57. Li, H., Zhao, R. & Wang, X. *Highly Efficient Forward and Backward Propagation of Convolutional Neural Networks for Pixelwise Classification*. arXiv preprint [arXiv:1412.4526](https://arxiv.org/abs/1412.4526) (2014).
58. Farabet, C., Couprie, C., Najman, L. & LeCun, Y. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1915–1929 (2012).
59. Hamwood, J., Alonso-Caneiro, D., Read, S. A., Vincent, S. J. & Collins, M. J. Effect of patch size and network architecture on a convolutional neural network approach for automatic segmentation of oct retinal layers. *Biomed. Opt. Express* **9**, 3049–3066 (2018).
60. Kotsiantis, S. *et al.* Handling imbalanced datasets: A review. *GESTS Int. Trans. Comput. Sci. Eng.* **30**, 25–36 (2006).
61. Wang, G., Kang, W., Wu, Q., Wang, Z. & Gao, J. Generative adversarial network (gan) based data augmentation for palmprint recognition. In *2018 Digital Image Computing: Techniques and Applications (DICTA)* 1–7 (2018).
62. Zhang, D. *et al.* *Deep Learning for Medical Image Segmentation: Tricks, Challenges and Future Directions*. arXiv preprint [arXiv:2209.10307](https://arxiv.org/abs/2209.10307) (2022).
63. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32** (2019).

64. Cuda toolkit. <https://developer.nvidia.com/cuda-toolkit>. Accessed: 06 Oct 2023.
65. Medical open network for artificial intelligence. <https://monai.io/>. Accessed: 06 Oct 2023.
66. Kavur, A. E. *et al.* Chaos challenge-combined (ct-mr) healthy abdominal organ segmentation. *Med. Image Anal.* **69**, 101950 (2021).
67. Shin, H.-C. *et al.* Medical image synthesis for data augmentation and anonymization using generative adversarial networks. In *Simulation and Synthesis in Medical Imaging: Third International Workshop, SASHIMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings* 3 1–11 (2018).

Acknowledgements

This research was funded by the Agence Nationale de la Recherche, grant number ANR-21-CE45-0002. This work was granted access to the HPC resources of IDRIS under the allocation 2023-AD010313884 made by GENCI.

Author contributions

F.Q., J.-L.A., F.M. and B.P. conceived the experiment(s); R.P., G.N., O.L., J.P., O.C. and J.-M.V. cured the data; F.Q., R.P., B.P., S.L., F.M., J.-M.V. and J.-L.A. wrote and edited the manuscript; B.P., S.L., J.-L.A., F.M., J.-M.V. and R.P. supervised the experiments; F.Q. and B.P. conducted and analysed the results. All authors reviewed the manuscript. F.Q. prepared all the figures.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-024-53528-9>.

Correspondence and requests for materials should be addressed to F.Q.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024