Efficient Algorithms for Medical Image Segmentation

José Martinho
Computer Science and Engineering Department
Instituto Superior Técnico
Lisboa, Portugal
jose.martinho@tecnico.ulisboa.pt

***Abstract*—With the growth in cancer cases and the increasing expenditures in the healthcare system, it is necessary to automate processes, aiming for a faster diagnostic and decrease in expenses. Although current technologies enable to capture high-resolution 3D images of organs, manual segmentation of organs and tumours is still a complex process that requires high expertise.**

State-of-the-art algorithms are already very accurate. However, they are very compute-intensive tasks, leading to the need for expensive hardware and energy wasting. Coupling state-of-the-art efficient feature extraction algorithms to the *nnUNet* segmentation framework, this work proposes novel efficient architectures for medical image segmentation. For some tasks, similar results were achieved using around 30% less coputing operations than the baseline *nnUNet*, also decreasing the inference time. Morevover, a better performance then *nnUNet* was achieved using architectures with slightly longer inference time.

***Index Terms*—Deep Learning, Medical Imaging, Segmentation, Efficiency**

I. Introduction

Malignant tumours are a relatively growing cause of death in Portugal (Figure 1), being the second greatest in this country. In fact, data from the World Health Organization (WHO), indicates that 1 in 6 world deaths is due to cancer [1].

Despite the advances in medicine and the increase in cancer survival rate [2], [3], deaths caused by cancer do not stop growing each year, and represent now about 1/4 of deaths in Portugal according to INE (Portuguese Statistics Institute). Trivially, we can conclude that the number of cancer cases is growing. This growth is mainly explained by the ageing population [4]. This demographic problem is making health expenses grow each year and cancer treatment represents around 6% of the total Portuguese Health Service expenditure [5].

Early diagnosis of tumours plays a significant role in the treatment of cancer and increases the survival rate of patients [6]. To this end, medical images should be acquired through radiological means and analyzed, with the goal of extracting information about the clinical situation of the patient. Cancer screening images are generally acquired through a computational process called CT (Computed Tomography). One of the most essential steps in acquiring valid information from these medical images is image segmentation. Since the segmentation of tumour areas is a truly specialized and time-consuming task requiring a fair amount of expertise, the automation of this process is considered advantageous.

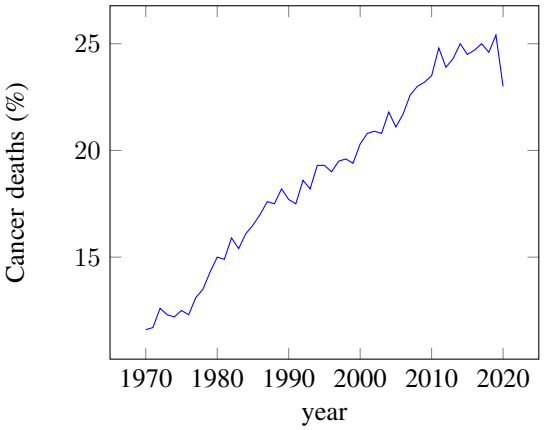


Fig. 1. Percentage of deaths caused by cancer, in Portugal. Data by PORDATA©

This field of computer vision is in extensive development and the advances are evident every year. However, frequently, improvement is related to the increase of computational resources needed to evaluate each image. These computing operations, apart from representing excessive energy consumption, are not always available for health institutions.

Using state-of-the-art computationally efficient feature extraction algorithms, the goal of this work is to try to improve a baseline encoder-decoder network (*U-Net*), by comparing the performance and resource consumption of different encoder-decoder combinations. The chosen dataset is *KiTS*, an open database of kidney cancer CTs and annotations done by specialists. Kidney tumours are among the top 10 most frequent cancers in men [3] and have a survival rate near the average cancer [2]. The growing trend follows the general malignant tumour growth already covered in this section.

II. Related Work

A. Convolutional Neural Networks

CNNs (Convolutional Neural Networks) are a specific type of deep learning algorithms targeted at spatially structured data, such as images. Quickly these algorithms turned into the state-of-the-art architecture family for visual tasks. Trying to simulate the visual perception process of living beings, CNNs learn small details during the first stages and use

those details to learn more high-level features and output valid information, such as segmentation, classification or other semantic characteristics of the input. Inspired by Fukushima's Neocognitron [7] that emerged in 1980, the first scientists to introduce the concept of CNNs were Lecun *et al.* [8] in 1998, with their acclaimed *LeNet*. The name makes reference to the mathematical convolution operation, the basis for the feature extraction operations used by these networks. In general, CNNs are composed of various convolutional layers, alternated with pooling layers (to reduce the complexity of the network) and non-linear activation functions. Convolutional layers comprehend various kernels, which will learn different features of their input. This operation may be seen as the weighted sum of neighbour pixels of an image, whose weights are the kernel and are learned through back-propagation.

More than a decade later, with the increase of the computational power of GPUs (General Processing Units), one of the first GPU-based CNNs used in such tasks was presented by Krizhevsky *et al.* [9], making them win the ILSVRC (ImageNet Large Scale Visual Recognition Challenge). Their proposed *AlexNet* consisted of a sequence of 5 convolutional and 3 fully connected layers. The main particularity of their approach is the parallelization technique used. Aiming to be able to run the model in low memory GPUs, they divided the kernels between 2 processing units, which shared outputs among them only on certain layers.

The increasing of computational power of existent hardware paved the way for the emergence of more and more complex algorithms. This complexity, however, brings some drawbacks: besides the need for powerful hardware and the increase in inference and training time, the increase in complexity does not always mean an increase in performance. Hence, scientific works on CNNs started to make an effort to amortize these hitches.

In 2015, Szegedy *et al.* [10] introduced the Inception modules. These modules consisted of a set of convolutions with different kernel sizes, whose outputs were concatenated with others', in order to extract features at different dimensions and scale without blowing up in computational complexity. The developed network, *GoogLeNet*, made up of these modules, was able to perform state-of-the-art results on ILSVRC 2014 with a top-5 error rate of 6.67%.

Comparing the performance of shallow with deeper networks, He *et al.* [11] realised that shallow networks may perform better than their deeper counterparts because deeper layers hardly learn identity mapping. Aiming to combat this degradation problem, they introduced in 2016 the *ResNet*, a remarkable deep learning framework that learns residual functions. Outputs from early layers are directly summed to the outputs from deeper layers (residual), through *residual connections*. In extreme cases, the residual would be pushed to zero, which proved to be easier to learn than identity mapping. This technique allowed to build a 152-layer architecture, which achieved state-of-the-art performance on the *ImageNet* dataset, with less resources than previous approaches [10].

Aiming to further exploring residual learning, Huang *et al.* [12] proposed in 2017 the *DenseNet*. This architecture makes use of residual connections to connect every layer to every following layer. This approach proved to reduce the vanishing gradient problem, strengthen feature propagation and also reduce the number of parameters. This architecture enabled the researchers to achieve state-of-the-art performance on CIFAR, SVHN and ILSVRC datasets while using a smaller number of parameters than *ResNet* [11].

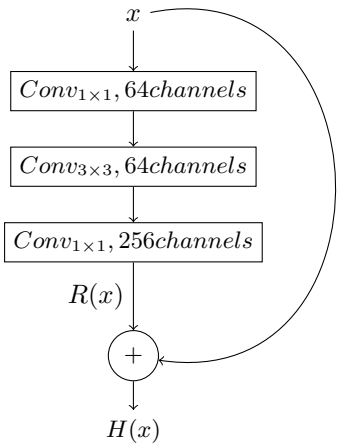


Fig. 2. Building block of the *ResNet* architecture. The input is summed to the output and stacked layers aim to learn the residual function $R(x) = H(x) - x$

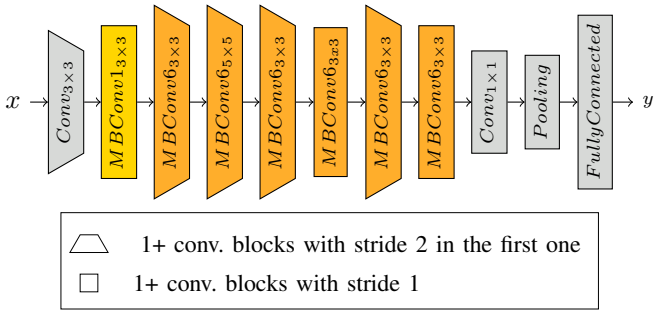


Fig. 3. *EfficientNet* architecture. This network comprises 9 phases, that consist of a first convolution, followed by 7 *MBConv* blocks and a final mapping phase. The number of repetitions of each *MBConv* block, their number of channels and the input resolution are scaled following the method proposed by the authors [13]

Recently, in 2020, motivated by the need of creating scaling criteria for CNNs, Tan *et al.* [13] presented the *EfficientNet*. The proposed method combines width (number of channels), depth (number of convolutions) and resolution scaling to adapt the size of the networks to the usage requirements. Besides applying this method to existing architectures (such as the *MobileNets* [14], [15] and ResNet [11]), authors have also proposed their own series of CNNs (a baseline and its scaled variations), which achieved state-of-the-art results on the *ImageNet* dataset, using at least 4 times fewer FLOPs and 2 times fewer parameters than previous state-of-the-art architectures. Heavily influenced by the *MobileNets* [14], [15],

this architecture comprises an initial 3×3 convolution used to expand the number of channels, followed by 7 scalable phases composed of *MBConv* blocks, and the final output phase (Figure 3). The resolution, number of channels and number of convolutions used in each *MBconv* phase are determined by the scale factor, which varies among *EfficientNet* variations.

MBConvs are convolutional blocks originally presented by Sandler *et al.* [15], characterized by their inverted residual shape. Each block comprises an optional 1×1 convolution to increase the number of channels, followed by a depthwise convolution, a *Squeeze-Excitation* module [16] and a final convolution whose output is then added to the input of the block, as with the *Resnet* [11]. Every convolutional operation is followed by a batch normalization layer and a swish [17] activation function.

Later, in 2021, the same authors [18] proposed an improved model of this network, *EfficientNetv2*. The main architectural difference relative to the former is the use of *Fused-MBConv*s, which replaces the first expanding convolution and the depthwise convolution with an expanding 3x3 convolution.

B. CNN Architectures for Image Segmentation

Image segmentation is the process of assigning a class to each pixel of an image, creating different regions of pixels that ideally correspond to different objects or different classes of objects. There are different types of image segmentation: semantic segmentation (where objects of the same class are assigned the same label), instance segmentation (where different objects of the same class are assigned different labels), and panoptic segmentation (a combination of the previous two). Image segmentation has umpteen use cases, such as autonomous vehicles, medical image analysis and digital marketing [19], making it a very important problem in the computer vision field. Following the success of CNNs on image feature extraction, and motivated by the relevancy of image segmentation in some fields of science such as medicine, this technique rapidly become standard for tasks of this kind.

In 2012, Ciresan *et al.* [20] proposed a "classical" CNN, composed of convolutional and max pooling layers, followed by several fully connected layers. The inference and training were done to the neighbouring region (patch) of a pixel and the output represented the class that each pixel belongs to, from among 2 classes. Although disruptive, this network has shown some drawbacks, such as slow "patch-by-patch" inference and the trade-off between good context when using larger patches and good localization accuracy with smaller ones [21].

Aiming to increase the segmentation accuracy of this method, the first FCNs (Fully Convolutional Layers) have surfaced. Long *et al.* [22] propose to make the "convolutionalization" of existing classification network architectures, such as *AlexNet* [9], *VGG16* [23] and *GoogLeNet* [10], removing fully-connected layers and replacing them with convolutions. This makes the networks resolution-agnostic and enables them to output a spatial output map, making them fit for segmentation problems. Moreover, the computation is highly amortized over the overlapping regions of input patches. Although this method works, the results were not satisfactory, which motivated the same authors to propose a novel network architecture, that introduced skip connections to combine coarse with finer information.

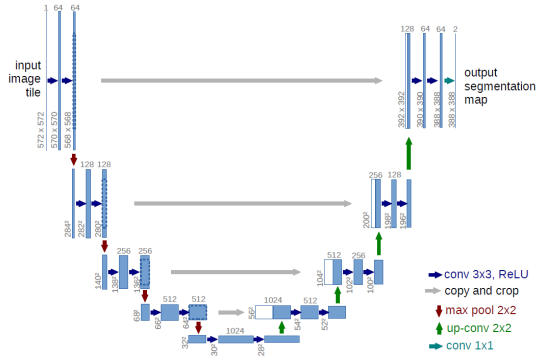


Fig. 4. The *U-Net* architecture consists of a contractive and an expansive path, where feature maps' dimensions are halved and doubled respectively, while the number of feature maps increases in the former and decreases in the latter. Also, the contractive path's feature maps are concatenated with their expansive path counterpart to recover spatial information.

Based on the work on FCNs [22] and other scientific papers that suggested to combine finer with coarser feature maps to produce the output [24], [25], Ronneberger *et al.* [21] came up with one of the most relevant works on image segmentation, the *U-Net* (Figure 4). Originally conceived to be applied to biomedical imaging, it is now the basis for the vast majority of state-of-the-art works on semantic segmentation. This architecture consists of two branches: a contractive and an expansive path. The former (also known as the encoder) follows a typical convolutional network architecture. It is a sequence of blocks of two convolutions followed by max pooling for down-sampling. At each down-sampling step, the number of feature maps is doubled. The expansive path (also known as the decoder) consists of the inverse operations: transposed convolutions to perform the up-sampling, followed by convolutions used to halve the feature map size. At the end of the decoder, a 1×1 convolution is applied in order to map each pixel to the desired class. At each level of the encoder, there is a "skip connection" to the corresponding decoder resolution step. These connections consist of the concatenation of the feature maps in the contraction path to the corresponding ones in the up-sampling path. This process enables the decoder to recover the spatial information that may have been lost during the contraction step.

Aiming to overcome data scarcity specific of the biomedical domain, the *U-Net* applies data augmentation, specifically random elastic deformations, and makes use of a dropout layer at the end of the contracting path. With the same goal, other authors introduced different error functions, to handle class imbalance (background is often more frequent than foreground labels). Çiçek *et al.* [26] propose a weighted cross entropy error, whereas Milletari *et al.* [27] suggests to

maximize the Dice Coefficient Function. Other approaches apply a combination of both binary cross-entropy loss and dice coefficient [28] [29].

The encoder-decoder model used by the *U-net* has been the target of extensive studies and improvements. The need of performing segmentation on 3D images led to 3D variations of the *U-Net* [26], [27]. Specifically, the *V-net* [27] uses residual connections in the encoder convolutions, improving learning speed and achieving better results than state-of-the-art architectures. In 2018, Oktay, Schlemper, Folgoc, *et al.* introduced attention gates (AG) to the *U-Net*'s skip connections [30], aiming to suppress feature responses in irrelevant background regions, in order to reduce false positives. This latter technique has showed some improvements in the expanse of increased computational requirements. Another relevant work was the UNet+++ [28], a *U-Net*-like architecture, but with an expansive path for every encoder level, enabling deep supervision to be used on every full-resolution feature map. Different encoders and decoders were also tested on these architectures, such as *DenseNets* [31] or *EfficientNets* [32].

C. Medical Image Segmentation

Compared to natural images, medical images require a much greater level of accuracy. Otherwise, automatic segmentation can lead to poor user experience [28]. Medical datasets feature many characteristics that differentiate them from other datasets, such as the low number of classes [33], data scarcity [34] or class imbalance [27]. Other factors make these tasks challenging, such as the variation in the appearance of certain organs and medical images and the pollution of medical images with artefacts and distortions [27].

The *U-Net* has undoubtedly played a crucial role in medical imaging segmentation. We can clearly note this by looking at the leaderboard of the 2019 edition of KiTS, one of the biggest segmentation challenges hosted by the MICCAI: all the 15-top methods are *U-Net* like architectures [29].

1) State-of-the-Art Tumor Segmentation Techniques: In 2018, the second place on the *BraTS* (Brain Tumor Segmentation) Challenge's leaderboard went to a CNN model proposed by Isensee, Kickingereder, Wick, *et al.*, called No New-Net [35]. This model assumes that a well trained *U-Net* or 3D *U-Net* [21], [26] is more dataset agnostic and may show better results than other *U-Net* variations (such as residual connections [11], [27], dense connections [12], [31] or attention gates [30]). Later on, the same authors proposed the *nnU-net* [29], a self configuring deep learning method for medical image segmentation. The self-configuration enables less experienced users to train the network without great knowledge and ensures the best approach for each dataset. The key idea behind this approach is to capture the *dataset fingerprint*, which describes the dataset used for training and to elaborate the *pipeline fingerprint* based on it (Figure 5).

The *pipeline fingerprint* consists of 3 types of parameters: *Blueprint Parameters*, *Inferred Parameters* and *Empirical Parameters*. *Blueprint Parameters* are key network choices, that won't change among different datasets. It features key

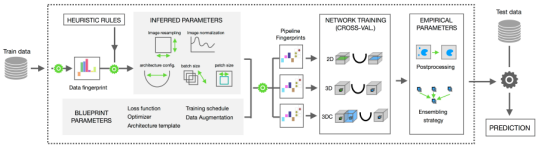


Fig. 5. The *nnU-Net*'s parameters consists of Inferred Parameters (inferred through the data fingerprint), Blueprint Parameters (key choices) and Empirical Parameters (chosen by cross-validation)

details about the network architecture and training parameters. *Inferred Parameters* are variable depending on the dataset. On data preprocessing, they specify the target spacing desired for the training samples and normalization and resampling techniques to be applied to the training set. They also adjust batch and patch size to fit hardware limitations, as well as the number of downsampling/upsampling steps to be performed by the encoder/decoder. A 2D *U-Net*, a 3D *U-Net* and a cascade 3D *U-Net* are then trained and the best ensemble is chosen by cross-validation. This ensemble constitutes the *Empirical Parameters*. For training, the authors propose deep supervision at every but the two lowest resolutions and as a loss function, the sum of cross-entropy and Dice loss is proposed.

nnU-net proved to be very impactful on medical imaging segmentation research. Many applications and variations of this approach have been proposed both by the same authors as well as by other researchers.

A large portion of *KiTS* 2019 and 2021's approaches is based on the successful *nnU-Net* and its variants. Hou *et al.* [36] propose to use the *nnU-Net* on a 3 stage approach: after the pre-processing, first and second stage use a *nnU-Net* in order to localize and segment the kidney. The tumour segmentation is then performed using a custom-made 3D *U-Net*. Other similar 2-stage approaches have also been followed [37], [38], where they take advantage of 2 *nnU-Nets* in order to first localize the kidneys and then segment the tumour. The approach from Zhao *et al.* [39] goes even further, using 4 distinct *nnU-Net*s: one for firstly segment the RoI (Region of Interest) of the kidney and then the other 3 to finely segment the kidney, the tumour and the mass, respectively. The latter 2 receive as input both the RoI and the finely segmented kidney. The authors also propose a novel loss function, the Surface Dice Loss, based on the Surface Dice Coefficient. This loss penalizes the model based on the distance between the wrongly classified pixels and the boundary of the ground-truth region. Golts, Khapun, Shats, *et al.* [40] also use the *nnU-Net* architecture, and propose a loss function that penalizes the output of a pixel based on its neighbouring pixels. On the other hand, Yang *et al.* [41] propose to train a model on a large medical imaging dataset and then use the best weights to initialize the training on the *KiTS* dataset. All the datasets are pre-processed with the methods suggested by the *nnU-Net* and the model used for training is a *U-Net* with residual connections.

Despite the predominance of the use of *nnU-Net* by the top places in this challenge, other promising approaches are

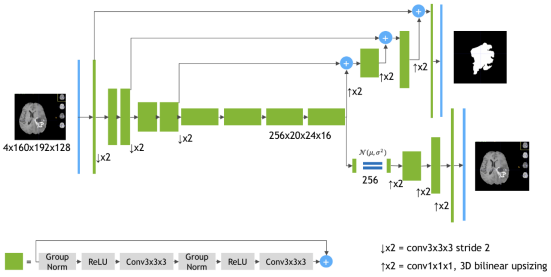


Fig. 6. Architecture proposed by Myronenko [43]. Two different decoders are attached at the end of the encoder: One for performing the segmentation and a VAE branch for "regularizing" encoder training.

worth noting. Myronenko *et al.* [42] presented an encoder-decoder architecture consisting of a larger encoder and a smaller decoder, with a series of convolutions and residual connections. The output of the decoder is concatenated with the output of a parallel boundary stream, which consists of a series of convolutions and attention gates, similar to the approach proposed by Oktay *et al.* [30].

In 2019, the same author proposed another asymmetrical encoder-decoder architecture with two parallel decoders. Here, besides UNet's classic decoder that outputs the segmentation mask, an auxiliary VAE (Variational Auto-Encoder) decoder is implemented in the other branch and used during training only, trying to reconstruct the original, "regularizing" the learning process of the encoder (Figure 6). The novel loss function presented is a weighted sum of VAE loss and segmentation loss and is given by:

$$L = L_{dice} + 0.1 \times L_{L2} + 0.1 L_{KL} \tag{1}$$

where L_{dice} is the dice loss of the output segmentation, L_{L2} is the $L2$ loss of the VAE output and L_{KL} is the KL divergence between the estimated distribution $\mathcal{N}(\mu, \sigma^2)$ and a prior distribution $\mathcal{N}(0, 1)$. This approach has won the first place in *BraTS* 2018, outperforming the original *nnU-Net* [35]. Jiang *et al.* [44] followed a similar strategy to win *BraTS* in 2019, with a two-stage segmentation approach, where the first stage is performed by a *U-Net* with a larger encoder and the second stage is performed by a similar network, but with a double decoder, one of which is only used during training. This way, the loss function and the gradient can be applied based on 2 decoders instead of just one.

Similarly to the approaches by Myronenko and Jiang *et al.*, also the approach presented by Wang *et al.* [45] makes use of 2 parallel branches on a *U-Net* like architecture. However, the latter makes use of both for inference and not just one as the others do. The several input channels are divided among the two branches and a series of connections are made between both of them. This approach was awarded second place in *BraTS* 2020 challenge, only beaten by a variation of the *nnU-Net* [46] presented by the original authors. This variation focuses especially on data augmentation and an increase in training batch size.

Aside from U-Net architectures, other approaches have been proved relevant for Brain Tumor Segmentation among them the H^2NF-*Net* [47]. This 2 stage approach makes use of 2 networks composed of several fully connected residual networks to process the input images on several resolutions. The outputs of the final stage are then merged using a special module. This has also been awarded second place in *BraTS* 2020 competition. Also McKinley *et al.* [48] proposed a non-*U-Net* architecture, making use of residual connections and Attention Gates. One of the interesting characteristics of this approach is the outputs of the network: besides the pixels' classes prediction, it also outputs the prediction that a certain classification disagrees with the Ground Truth. This enables the formulation of a novel loss function, that combines the output classification, the output disagreement probability and the ground truth.

2) Discussion: Encoder-decoder architectures with skip connections (such as the *U-Net*) are dominant in the image segmentation field. The use of extra skip connections [28], [49] or auxiliary decoder branches [42]–[45] have been widely studied and the results have been satisfactory. Different encoders have also been proposed based on state-of-the-art convolutional frameworks, such as *ResNet* [27], [41] or *DenseNet* [12], [31]. However, *EfficientNet* has not been yet very explored in the encoder-decoder segmentation procedures, despite the good performance it shows on the original paper as a feature extraction network. To my knowledge, the only approach that used *EfficientNet* for medical image segmentation was *EfficientUNet++* [32], which has shown promising results. Other approaches make use of Attention Mechanisms [30], [48], [49], which also performed well.

nnU-Net is undoubtedly the most influential work on state-of-the-art biomedical image segmentation works. A vast majority of recent approaches make use of it either for pre-processing or as a baseline network for their architectures.

III. Architecture

Some works show that *U-Net* and similar architectures can be successfully modified to work with different feature extraction operations in the encoder path [12], [27], [31], [41]. Given that the *EfficientNet* family is currently one of the state-of-the-art image classification networks, and being it yet under-explored as an encoder in encoder-decoder architectures, during this work several modified *EfficientNet* and *EfficientNetV2* variations were extensively tested as an encoder in the *nnU-Net* framework. Moreover, given that this architectural family use an inferior number of computational resources in comparison to other state-of-the-art algorithms, it makes sense to apply it to high-resolution 3D images. Furthermore, a novel decoder was developed based on the *MBConv* blocks [13]–[15], [18].

This work focuses exclusively on the 3D model of the *nnU-Net* framework.

A. EfficientNet as an encoder

One of the challenges to integrating classification architectures into the *U-Net* is to divide them into resolution stages

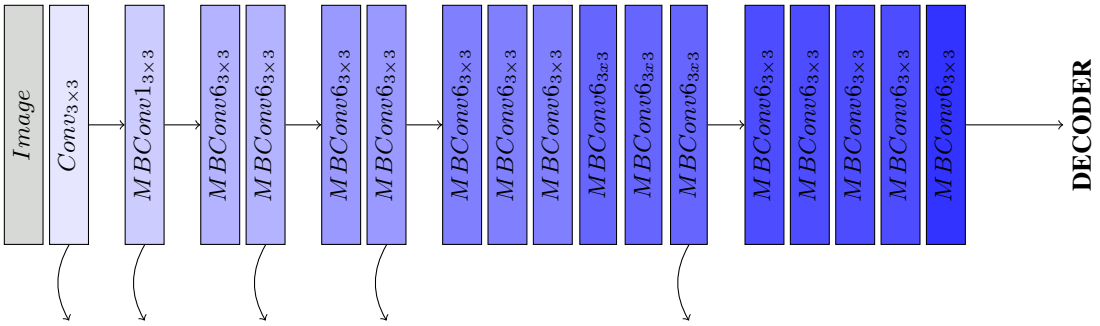


Fig. 7. *EfficientNet-b0* encoder. Skip connections are marked with a bent arrow. Different color tonality means different *EfficientNet* stage. Note that each encoder stage may contain convolutions from different *EfficientNet* stages.

so that they may be inserted into the encoder. To this end, some changes were made to the original proposed *EfficientNet* model. Firstly, the initial convolution proposed in the original *EfficentNet* publication suggests a fixed-size input strided convolution (whose input size changes among *EfficientNet* variations). However, as the first stage of the U-Net requires a feature map with the exact resolution as the input but wider (with more channels), the stride of this first convolution was changed to 1. Also, the input to the first convolution was made variation agnostic, and always fixed to the patch size. To compensate for the stride change, the first *MBConv* block is implemented with stride 2. The last stage from the *EfficientNet* is pruned, as we don't need to output a mapping, only the latent map created by the convolutions.

Despite the change of the encoder, some of the *nnU-Net*'s Inferred Parameters are followed. For the *KiTS* dataset, *nnU-Net* determines a patch size of $128 \times 128 \times 128$ and therefore it proposes to perform 5 downsamples in the encoder, as the rule of the *nnU-Net* is to perform downsamples until the shape is $4 \times 4 \times 4$. We can observe in Figure 7 that this verifies for the proposed encoder. As the *EfficientNet* family is invariant on the number of downsampling operations, this condition verifies for every encoder belonging to the family.

A similar approach was followed for the *EfficientNetV2*. This newer variant is very similar to the original *EfficientNet*, but makes use of *Fused-MBConv* blocks, already covered in section II.

B. Implementation of a novel decoder

Trying to further explore the powerful capacities of *MBConvs* [13]–[15], [18], a novel decoder was developed, replacing the generic convolutions proposed by the *nnU-Net* with *MBConvs*. These blocks are repeated twice each stage, such as proposed by the original *nnU-Net*, and have kernel sizes also defined by the *nnU-Net* inferred parameters. Squeeze-excitation is applied with a ratio similar to the one in the encoder (0.25) and the expanding ratio of the *MBConv* is 6.

As shown in Figure 8, the block from the deeper level of the encoder is upsampled using the transposed convolution proposed in the original *nnU-Net* implementation. It is important to note that the concatenated volume may not contain the same number of channels as the upsampled one: whereas the

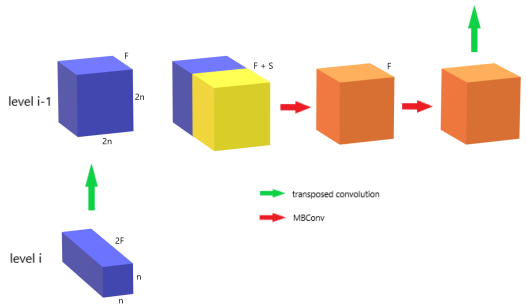


Fig. 8. The FullyEfficient decoder. The yellow block corresponds the concatenation from the skip connection. The transposed convolution remains unchanged from the *nnU-Net* implementation. *MBConvs* are the main building block.

original encoder follows the rule of duplicating the number of channels for each level, the *EfficientNet* does not follow a similar pattern. Therefore, besides the fact that the encoder and decoder are asymmetric in the number of channels, the block originated from the concatenation of the two varies in shape among different variations of the *EfficientNet* encoder. This new decoder was named *FullyEfficient-UNet*.

IV. Model Training Methodology

A. Overview

The *nnU-Net* paper [29] proposes to use batch learning with 5-fold cross-validation for training. However, due to the excessive training time, it was unfeasible to follow the same strategy. Also, as the ground-truth of the test set of the *KiTS* dataset is not publicly accessible, the training set was split into train, validation and test sets.

Every model was trained using an NVIDIA®Tesla®V100S with 32GB VRAM.

The loss function, hyperparameters and preprocessing are the same as the proposed by the *nnU-Net Blueprint* and *Inferred parameters* and will be addressed in the chapter.

B. Loss Function

As suggested in the *nnU-Net* original paper, the loss function that was used was the sum of Generalized Dice Loss and

Cross Entropy Loss:

$$L = L_{Dice} + L_{CE} \tag{2}$$

1) Soft Dice Loss: The DSC (Dice Similarity Coefficient) can be used to measure the similarity between the predicted segmentation masks of each class and the corresponding ground truth mask. As the output is a set of probabilities, and not the mask itself, the Soft Dice Metric can be used, where instead of using thresholding to get the predicted mask and intersect with the ground-truth mask, we can make use of the probabilities to make a weighted mask. Hence, the used metric is given by:

$$D = \frac{2TP}{2TP + FP + FN} = \frac{2\sum_n^N \hat{p}_n * y_n}{2\sum_n^N \hat{p}_n * y_n + \hat{p}_n * (1 - y_n) + (1 - \hat{p}_n) * y_n} \tag{3}$$

where $\hat{p}$ is the output probability matrix, y is the ground-truth and $*$ represents the element-wise multiplication operation. This metric is calculated for each class and then averaged.

Note that these coefficients are always between 0 and 1, with values close to 1 indicating that the predicted map is very close to the segmentation ground-truth. As the goal is to approximate the function's maximum, the loss function must be the negative coefficient:

$$L_{Dice} = -D \tag{4}$$

2) Cross-entropy Loss: As Dice loss may lose accuracy with batch-based learning and does not deal well with class oversampling [29], authors of *nnU-Net* empirically note that these hitches may be overcome by combining it with cross-entropy loss. This loss is used in many deep learning tasks and is given by:

$$L_{CE} = -\sum_n^N y_n \cdot \log \hat{p}_n \tag{5}$$

C. Hyperparameters

Most of the hyperparameters are decided on the *nnU-Net* pipeline, as explained in section II. During learning, each batch comprises two $128 \times 128 \times 128$ patches. Patches are randomly sampled from training cases, assuring that one-third of them contains foreground voxels.

nnU-Net also proposes to train the models for 1000 epochs. Each epoch comprises 250 training iterations, each composed of a forward and backward pass of a mini-batch. The learning rate is initialized at 0.01 (with a Nesterov momentum of 0.99) and decayed using the *polyLR* policy.

However, this strategy could not always be followed, as covered in the next chapter.

V. Experimental Results

A. Overview

As explained in section IV, *nnU-Net* proposes training during 1000 epochs with an initial learning rate of 0.01, which is decayed following the *polyLR* policy. However, some architectures were revealed to be untrainable with these settings, due to gradient explosion. The evaluation metrics are the ones proposed by the challenges that provide the datasets and these are the metrics used for comparison between different architectures.

B. Metrics

The metrics used for evaluation are the ones proposed by the organizations of the challenges the datasets were obtained from. These are the metrics used for ranking and the most fair for comparison among different models.

KiTS challenge proposes an evaluation that comprises separate evaluations of different foreground classes: kidney, cysts and tumours. However, these are not evaluated individually. Aiming to avoid double penalization in some cases, the evaluation is performed on hierarchical classes:

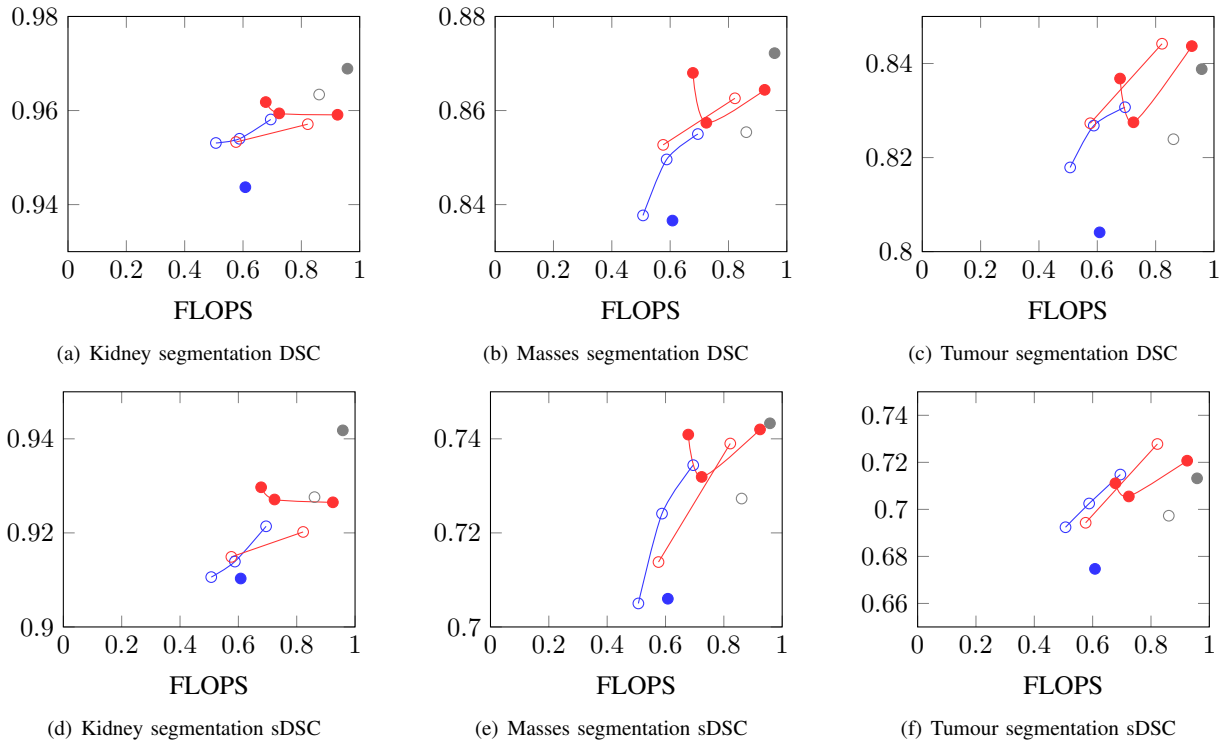
- Kidney and Masses (Kidney + Tumour + Cyst)
- Kidney Mass (Tumour + Cyst)
- Tumor

Each of these hierarchical classes is evaluated with the DSC (presented in section IV) and sDSC (surface Dice Similarity Coefficient) [50]. The latter firstly analyzes the multiple different segmentation groundtruths to calculate the "acceptable deviation", a distance that defines the limit to classify the predicted segmentation's surface as "acceptable or "unacceptable". This limit corresponds to the 95th percentile of the distances of those segmentations done by professionals. For *KiTS* this metric is viable, given that for each case there are 3 different annotations done by different professionals.

C. Performance Analysis

When training the *EfficientNet*-b1 and over encoders with the original *nnU-Net* decoder, the gradient exploded and training was unfeasible to be made with the riginal hyperparameters. Although decreasing the learning rate and increasing the number of epochs would be a naive way to solve this problem, these architectures were not evaluated due to time constraints and the need to evaluate every network with the same conditions. When plugging the novel *FullyEfficient* decoder with these encoders, however, the gradient would not explode and the training would become possible. Other networks were left aside also due to the time constraints of this work. The trained and tested architectures were *GenericUNet* (baseline), *FullyEfficient GenericUNet*, *EfficientUNet*-b0 , *EfficientUNetV2*-s, *EfficientUNetV2*-m, *EfficientUNetV2*-l, *FullyEfficientUNet*-b0, *FullyEfficientUNet*-b4, *FullyEfficientUNet*-b7, *FullyEfficientUNetV2*-s and *FullyEfficientUNetV2*-l.

As it is possible to notice in Figure 9, every trained architecture executes fewer FLOPs than the original *nnU-Net* network (depicted in a filled grey circle), maintaining



(a) Kidney segmentation DSC (b) Masses segmentation DSC (c) Tumour segmentation DSC

(d) Kidney segmentation sDSC (e) Masses segmentation sDSC (f) Tumour segmentation sDSC

Fig. 9. Comparison of DSC of different encoder-decoder combinations as a function of the number of the number of FLOPS. Architectures with the *EfficientNet* encoder are represented in blue, whereas architectures with the *EfficientNetV2* encoder are represented in red. Original encoder is depicted in gray. Filled circles and open circumferences represent the original and the novel decoder, respectively.

the results very similar: score metrics only vary at most 0.04 among different architectures.

For the whole kidney segmentation (9(a), 9(d)), the baseline network is clearly the most performative. Among more efficient architectures, *EfficientNetV2* is the family that performs the best. However, an odd event occurs with these encoders: smaller networks perform better than larger ones. This can be explained with overfitting, due to the great number of parameters these networks comprise, as shown in the section A.

Notably, the smaller the region to be segmented, the better the results of efficient architectures in comparison with the baseline. For masses segmentation (9(b), 9(e)) and tumour segmentation (9(c), 9(f)), *Efficient-UnetV2*-S shows very similar results to the baseline executing 40% less FLOPs. Moreover, *FullyEfficientUNetV2*-L shows better results than the baseline in the tumour segmentation task, with 18% less FLOPs.

The impact of the *FullyEfficient* decoder is not clear. In conjunction with the *EfficientNet* encoder, despite the lack of results with the original decoder due to the gradient problem previously approached, both *FullyEfficientNet*-b0 and *FullyEfficientNet*-b1 have shown better results than *EfficientNet*-b0 on almost all the tasks. However, for the *EfficientNetV2* encoder family results are not so clear: despite the novel decoder worsening results in the kidney segmentation task, the most performative architecture for tumour segmentation is the *FullyEfficientNetV2*-L.

In spite of the reduced number of FLOPs, efficient architectures comprise a large number of parameters. The largest networks from both the *EfficientNet* and *EfficientNetV2* families require even a greater number of parameters than the baseline. This is due to the fact that *MBConv* blocks have more parameters than original convolutional blocks and are repeated more times on each encoder level. Although a larger number of parameters may be a cause of overfitting, this phenomenon is not clearly visible, as networks with more parameters are able to achieve better results than the baseline on some tasks.

Another important metric is the inference time. Contrary to expectations, a reduction of the number of FLOPs was not directly linked to a reduction in the inference time. This fact could be related to the larger umber of covolutions introduced by the efficient architectures, which made the algorithm less parallelizable.

VI. Discussion

This work aimed to contribute to the efficiency of medical image segmentation algorithms. Most existing state-of-the-art approaches, although already very accurate, lack in efficiency, requiring a large number of FLOPs to execute and, consequently, a longer inference time. To achieve efficiency, multiple encoders and decoders have been comprehensively tested and attached to the *nnU-Net*, a popular biomedical image segmentation framework.

The architectures proposed in this work revealed to require less FLOPs than the original one, in spite of the fact that this

relation is not directly related to a decrease in inference time or in the number of parameters. In fact, some architectures with an *EfficientNet* backbone have shown to require more parameters than the original network, but this didn't lead to *overfit*, possibly due to the residual connections comprised in *MBConv*s and *Fused-MBConv*s. Actually, for the masses segmentation and tumour segmentation tasks, *EfficientUNetV2*-L and *FullyEfficientUNetV2*-L performed better than the original *nnU-Net* despite having more parameters and less FLOPs than the latter.

EfficientUnetV2-S, with 30% less FLOPs than the original *nnU-Net* and also a lower inference time is able to achieve very similar results to the baseline. For masses segmentation, this architecture achieves a DSC of 0.868 and a sDSC of 0.7409, very near to the 0.8722 and 0.7433 achieved by the baseline. For the tumour segmentation, *EfficientUnetV2*-S reaches a DSC of 0.8368 and a sDSC of 0.7111, against the 0.8388 and 0.7132 achieved by the baseline. The best score for the tumour segmentation task was achieved by the *FullyEfficientUNetV2*-L, with an achieved DSC of 0.8442 and sDSC of 0.7278. This architecture requires less FLOPs to run, despite its larger inference time.

For a future work, it would be important to validate the findings of this work on other datasets, as other segmentation tasks could may require improved efficiency. Also, more architecture variations should be addressed, such as the introduction of a VAE in parallel with the decoder to regularize encoder training [19]. Discarded networks for gradient explosion should also be addressed in the future, with a reduced learning rate.

References

[1] F. de Almeida Fernandes, "Cancro mata mais do que sida, malária e tuberculose juntas," *Diário de Notícias*, Feb. 4, 2021. [Online]. Available: https://www.dn.pt/sociedade/cancro-mata-mais-do-que-sida-malaria-e-tuberculose-juntas-13312922.html (visited on 02/04/2021).

[2] M. Quaresma, M. P. Coleman, and B. Rachet, "40-year trends in an index of survival for all cancers combined and survival adjusted for age and sex for each cancer in england and wales, 1971-2011: A population-based study," *The Lancet*, vol. 385, no. 9974, pp. 1206–1218, Mar. 2015. DOI: 10.1016/s0140-6736(14)61396-9. [Online]. Available: https://doi.org/10.1016/s0140-6736(14)61396-9.

[3] A. da Costa Miranda, A. Mayer-da-Silva, L. Glória, and C. Brito, "Registo oncológico nacional de todos os tumores na população residente em portugal, em 2018," 2020. [Online]. Available: https://ron.min-saude.pt/media/2196/2021-0518_publica%C3%A7%C3%A3o-ron_2018.pdf.

[4] *Risk factors: Age*, Mar. 2021. [Online]. Available: https://www.cancer.gov/about-cancer/causes-prevention/risk/age.

[5] J. M. Lopes, F. Rocha-Gonçalves, M. Borges, P. Redondo, and J. Laranja-Pontes, "Custo do tratamento do cancro em portugal," 2017. [Online]. Available: https://ecancer.org/en/journal/article/765-the-cost-of-cancer-treatment-in-portugal/pdf/pt.

[6] D. Crosby, S. Bhatia, K. M. Brindle, *et al.*, "Early detection of cancer," *Science*, vol. 375, no. 6586, Mar. 2022. DOI: 10.1126/science.aay9040. [Online]. Available: https://doi.org/10.1126/science.aay9040.

[7] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980. DOI: 10.1007/bf00344251. [Online]. Available: https://doi.org/10.1007/bf00344251.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. DOI: 10.1109/5.726791.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[10] C. Szegedy, W. Liu, Y. Jia, *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2015. DOI: 10.1109/cvpr.2015.7298594. [Online]. Available: https://doi.org/10.1109/cvpr.2015.7298594.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016. DOI: 10.1109/cvpr.2016.90. [Online]. Available: https://doi.org/10.1109/cvpr.2016.90.

[12] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jul. 2017. DOI: 10.1109/cvpr.2017.243. [Online]. Available: https://doi.org/10.1109/cvpr.2017.243.

[13] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *arXiv:1905.11946 [cs, stat]*, Sep. 11, 2020. arXiv: 1905.11946. [Online]. Available: http://arxiv.org/abs/1905.11946 (visited on 10/06/2021).

[14] A. G. Howard, M. Zhu, B. Chen, *et al.*, *Mobilenets: Efficient convolutional neural networks for mobile vision applications*, 2017. DOI: 10.48550/ARXIV.1704.04861. [Online]. Available: https://arxiv.org/abs/1704.04861.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, IEEE, Jun. 2018. DOI: 10.1109/cvpr.2018.00474. [Online]. Available: https://doi.org/10.1109/cvpr.2018.00474.

[16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2018. DOI: 10.1109/cvpr.2018.00745. [Online]. Available: https://doi.org/10.1109/cvpr.2018.00745.

[17] P. Ramachandran, B. Zoph, and Q. V. Le, *Searching for activation functions*, 2017. DOI: 10.48550/ARXIV.1710.05941. [Online]. Available: https://arxiv.org/abs/1710.05941.

[18] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," 2021. DOI: 10.48550/ARXIV.2104.00298. [Online]. Available: https://arxiv.org/abs/2104.00298.

[19] NVIDIA. "Image segmentation." (2021), [Online]. Available: https://catalog.ngc.nvidia.com/orgs/nvidia/collections/imagesegmentation (visited on 09/09/2022).

[20] D. Ciresan, A. Giusti, L. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25, Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/459a4ddcb586f24efd9395aa7662bc7c-Paper.pdf.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science*, Springer International Publishing, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28. [Online]. Available: https://doi.org/10.1007/978-3-319-24574-4_28.

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2015. DOI: 10.1109/cvpr.2015.7298965. [Online]. Available: https://doi.org/10.1109/cvpr.2015.7298965.

[23] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2014. DOI: 10.48550/ARXIV.1409.1556. [Online]. Available: https://arxiv.org/abs/1409.1556.

[24] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, *Hypercolumns for object segmentation and fine-grained localization*, 2014. DOI: 10.48550/ARXIV.1411.5752. [Online]. Available: https://arxiv.org/abs/1411.5752.

[25] M. Seyedhosseini, M. Sajjadi, and T. Tasdizen, "Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks," in *2013 IEEE International Conference on Computer Vision*, IEEE, Dec. 2013. DOI: 10.1109/iccv.2013.269. [Online]. Available: https://doi.org/10.1109/iccv.2013.269.

[26] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2016*, Springer International Publishing, 2016, pp. 424–432. DOI: 10.1007/978-3-319-46723-8_49. [Online]. Available: https://doi.org/10.1007/978-3-319-46723-8_49.

[27] F. Milletari, N. Navab, and S.-A. Ahmadi, *V-net: Fully convolutional neural networks for volumetric medical image segmentation*, 2016. arXiv: 1606.04797 [cs.CV].

[28] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Springer International Publishing, 2018, pp. 3–11. DOI: 10.1007/978-3-030-00889-5_1. [Online]. Available: https://doi.org/10.1007/978-3-030-00889-5_1.

[29] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-net: A self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, Feb. 2021, ISSN: 1548-7105. DOI: 10.1038/s41592-020-01008-z.

[30] O. Oktay, J. Schlemper, L. L. Folgoc, *et al.*, "Attention U-Net: Learning Where to Look for the Pancreas," *arXiv:1804.03999 [cs]*, May 2018, arXiv: 1804.03999. [Online]. Available: http://arxiv.org/abs/1804.03999 (visited on 12/14/2021).

[31] S. Jegou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional DenseNets for semantic segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Jul. 2017. DOI: 10.1109/cvprw.2017.156. [Online]. Available: https://doi.org/10.1109/cvprw.2017.156.

[32] J. Lourenço-Silva, M. N. Menezes, T. Rodrigues, B. Silva, F. J. Pinto, and A. L. Oliveira, "Encoder-decoder architectures for clinically relevant coronary artery segmentation," in *Computational Advances in Bio and Medical Sciences*, Springer International Publishing, 2022, pp. 63–78. DOI: 10.1007/978-3-031-17531-2_6. [Online]. Available: https://doi.org/10.1007/978-3-031-17531-2_6.

[33] T. J. Jun, J. Kweon, Y.-H. Kim, and D. Kim, "T-net: Nested encoder–decoder architecture for the main vessel segmentation in coronary angiography," *Neural Networks*, vol. 128, pp. 216–233, Aug. 2020, ISSN: 0893-6080. DOI: 10.1016/j.neunet.2020.05.002. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2020.05.002.

[34] D. C. Castro, I. Walker, and B. Glocker, "Causality matters in medical imaging," *Nature Communications*, vol. 11, no. 1, Jul. 2020. DOI: 10.1038/s41467-020-17478-w. [Online]. Available: https://doi.org/10.1038/s41467-020-17478-w.

[35] F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-Hein, "No new-net," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, Springer International Publishing, 2019, pp. 234–244. DOI: 10.1007/978-3-030-11726-9_21. [Online]. Available: https://doi.org/10.1007/978-3-030-11726-9_21.

[36] X. Hou, C. Xie, F. Li, and Y. Nan, "Cascaded Semantic Segmentation for Kidney and Tumor," in *Submissions to the 2019 Kidney Tumor Segmentation Challenge: KiTS19*, University of Minnesota Libraries Publishing, 2019. DOI: 10.24926/548719.002. [Online]. Available: https://kits.lib.umn.edu/cascaded-semantic-segmentation-for-kidney-and-tumor/ (visited on 12/21/2021).

[37] Y. Zhang, Y. Wang, F. Hou, *et al.*, "Cascaded volumetric convolutional network for kidney tumor segmentation from CT volumes," in *Submissions to the 2019 Kidney Tumor Segmentation Challenge: KiTS19*, University of Minnesota Libraries Publishing, 2019. DOI: 10.24926/548719.004. [Online]. Available: https://doi.org/10.24926/548719.004.

[38] Y. George, "A coarse-to-fine 3d u-net network for semantic segmentation of kidney CT scans," in *Lecture Notes in Computer Science*, Springer International Publishing, 2022, pp. 137–142. DOI: 10.1007/978-3-030-98385-7_18. [Online]. Available: https://doi.org/10.1007/978-3-030-98385-7_18.

[39] Z. Zhao, H. Chen, and L. Wang, "A coarse-to-fine framework for the 2021 kidney and kidney tumor segmentation challenge," in *Lecture Notes in Computer Science*, Springer International Publishing, 2022, pp. 53–58. DOI: 10.1007/978-3-030-98385-7_8. [Online]. Available: https://doi.org/10.1007/978-3-030-98385-7_8.

[40] A. Golts, D. Khapun, D. Shats, Y. Shoshan, and F. Gilboa-Solomon, "An ensemble of 3d u-net based models for segmentation of kidney and masses in CT scans," in *Lecture Notes in Computer Science*, Springer International Publishing, 2022, pp. 103–115. DOI: 10.1007/978-3-030-98385-7_14. [Online]. Available: https://doi.org/10.1007/978-3-030-98385-7_14.

[41] X. Yang, J. Zhang, J. Zhang, and Y. Xia, "Transfer learning for KiTS21 challenge," in *Lecture Notes in Computer Science*, Springer International Publishing, 2022, pp. 158–163. DOI: 10.1007/978-3-030-98385-7_21. [Online]. Available: https://doi.org/10.1007/978-3-030-98385-7_21.

[42] A. Myronenko and A. Hatamizadeh, "3D Kidneys and Kidney Tumor Semantic Segmentation using Boundary-Aware Networks," *arXiv:1909.06684 [cs, eess]*, Sep. 2019, arXiv: 1909.06684. [Online]. Available: http://arxiv.org/abs/1909.06684 (visited on 12/21/2021).

[43] A. Myronenko, "3d MRI brain tumor segmentation using autoencoder regularization," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, Springer International Publishing, 2019, pp. 311–320. DOI: 10.1007/978-3-030-11726-9_28. [Online]. Available: https://doi.org/10.1007978-3-030-11726-9_28.

[44] Z. Jiang, C. Ding, M. Liu, and D. Tao, "Two-Stage Cascaded U-Net: 1st Place Solution to BraTS Challenge 2019 Segmentation Task," en, in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, A. Crimi and S. Bakas, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 231–241, ISBN: 978-3-030-46640-4. DOI: 10.1007/978-3-030-46640-4_22.

[45] Y. Wang, Y. Zhang, F. Hou, *et al.*, "Modality-pairing learning for brain tumor segmentation," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, Springer International Publishing, 2021, pp. 230–240. DOI: 10.1007/978-3-030-72084-1_21. [Online]. Available: https://doi.org/10.1007/978-3-030-72084-1_21.

[46] F. Isensee, P. F. Jäger, P. M. Full, P. Vollmuth, and K. H. Maier-Hein, "nnU-net for brain tumor segmentation," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, Springer International Publishing, 2021, pp. 118–132. DOI: 10.1007/978-3-030-72087-2_11. [Online]. Available: https://doi.org/10.1007/978-3-030-72087-2_11.

[47] H. Jia, W. Cai, H. Huang, and Y. Xia, "H2nf-net for brain tumor segmentation using multimodal MR imaging: 2nd place solution to BraTS challenge 2020 segmentation task," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, Springer International Publishing, 2021, pp. 58–68. DOI: 10.1007/978-3-030-72087-2_6. [Online]. Available: https://doi.org/10.1007/978-3-030-72087-2_6.

[48] R. McKinley, M. Rebsamen, R. Meier, and R. Wiest, "Triplanar Ensemble of 3D-to-2D CNNs with Label-Uncertainty for Brain Tumor Segmentation," en, in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, A. Crimi and S. Bakas, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 379–387, ISBN: 978-3-030-46640-4. DOI: 10.1007/978-3-030-46640-4_36.

[49] Y. Yuan, "Automatic brain tumor segmentation with scale attention network," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, Springer International Publishing, 2021, pp. 285–294. DOI: 10.1007/978-3-030-72084-1_26. [Online]. Available: https://doi.org/10.1007/978-3-030-72084-1_26.

[50] S. Nikolov, S. Blackwell, A. Zverovitch, *et al.*, *Deep learning to achieve clinically applicable segmentation of head and neck anatomy for radiotherapy*, 2018. DOI: 10.48550/ARXIV.1809.04430. [Online]. Available: https://arxiv.org/abs/1809.04430.

APPENDIX A
COMPLETE COMPARISON OF ARCHITECTURES

This appendix contains the complete comparison of the different networks on the *KiTS* dataset. The FullyEfficientUNetV2-L and EfficientUNetV2-L are more performative than the baselineon the tumour segmentation task, despite the lower number of FLOPs required.

Name	# FLOPs			# Parameters			Inference Time	Performance (DSC)			Performance (sDSC)		
Network	Network	Enc.	Dec.	Network	Enc.	Dec.	Network	Kidney	Masses	Tumour	Kidney	Masses	Tumour
GenericUNet	0.958T	0.279T	0.663T	31.196M	14.025M	15.349M	102.09s	**0.9689**	**0.8722**	0.8388	**0.9418**	**0.7433**	0.7132
FullyEffGenericUNet	0.861T	0.279T	0.565T	28.246M	14.025M	12.399M	152.07s	0.9634	0.8554	0.8239	0.9276	0.7273	0.6973
EfficientUNet-b0	0.608T	9.658G	0.581T	19.408M	5.969M	11.617M	81.79s	0.9437	0.8366	0.8041	0.9103	0.706	0.6747
EfficientUNet-b1	0.61T	11.926G	0.581T	22.165M	8.727M	11.617M	-	-	-	-	-	-	-
EfficientUNet-b2	0.611T	12.503G	0.582T	23.768M	10.124M	11.741M	-	-	-	-	-	-	-
EfficientUNet-b3	0.655T	17.916G	0.62T	27.509M	13.595M	11.928M	-	-	-	-	-	-	-
EfficientUNet-b4	0.69T	23.469G	0.65T	35.753M	21.406M	12.198M	-	-	-	-	-	-	-
EfficientUNet-b5	0.701T	32.024G	0.652T	47.829M	33.097M	12.419M	-	-	-	-	-	-	-
EfficientUNet-b6	0.749T	42.989G	0.689T	61.875M	46.697M	12.702M	-	-	-	-	-	-	-
EfficientUNet-b7	0.797T	59.897G	0.721T	86.96M	71.32M	12.999M	-	-	-	-	-	-	-
FullyEfficientUNet-b0	0.507T	9.658G	0.481T	15.201M	5.969M	7.41M	119.54s	0.9531	0.8377	0.8179	0.9106	0.705	0.6924
FullyEfficientUNet-b1	0.509T	11.926G	0.481T	17.958M	8.727M	7.41M	-	-	-	-	-	-	-
FullyEfficientUNet-b2	0.51T	12.503G	0.481T	19.574M	10.124M	7.547M	-	-	-	-	-	-	-
FullyEfficientUNet-b3	0.553T	17.916G	0.519T	23.344M	13.595M	7.763M	-	-	-	-	-	-	-
FullyEfficientUNet-b4	0.69T	23.469G	0.65T	35.753M	21.406M	12.198M	145.54s	0.954	0.8496	0.8268	0.9139	0.7241	0.7025
FullyEfficientUNet-b5	0.599T	32.024G	0.55T	43.756M	33.097M	8.346M	-	-	-	-	-	-	-
FullyEfficientUNet-b6	0.647T	42.989G	0.587T	57.874M	46.697M	8.702M	-	-	-	-	-	-	-
FullyEfficientUNet-b7	0.695T	59.897G	0.618T	83.044M	71.32M	9.083M	184.79s	0.9581	0.855	0.8307	0.9214	0.7344	0.7148
EfficientUNetV2-S	0.678T	94.991G	0.567T	38.267M	24.321M	12.287M	92.094s	0.9618	0.868	0.8368	0.9297	0.7409	0.7111
EfficientUNetV2-M	0.724T	0.139T	0.568T	74.065M	59.216M	12.536M	106.66	0.9594	0.8574	0.8275	0.9271	0.7319	0.7055
EfficientUNetV2-L	0.924T	0.327T	0.58T	0.143G	0.128G	13.131M	129.44s	0.9591	0.8644	0.8437	0.9265	0.742	0.7207
FullyEfficientUNetV2-S	0.576T	94.991G	0.465T	34.163M	24.321M	8.183M	133.48s	0.9533	0.8527	0.8273	0.9149	0.7138	0.6943
FullyEfficientUNetV2-M	0.622T	0.139T	0.466T	70.008M	59.216M	8.48M	-	-	-	-	-	-	-
FullyEfficientUNetV2-L	0.822T	0.327T	0.478T	0.139G	0.128G	9.244M	170.12s	0.9571	0.8626	**0.8442**	0.9202	0.739	**0.7278**

TABLE I
COMPLETE COMPARISON OF DIFFERENT NETWORKS. MISSING DATA IS DUE TO LACK OF TIME OR GRADIENT EXPLOSION. EVERY NETWORK WAS TRAINED DURING 1000 EPOCHS WITH AN INITIAL TRAINING RATE OF 0.01.