Building a Convolutional Network for Brain Tumor Segmentation

Giovanni Scognamiglio
g.scognamiglio2@studenti.unipi.it
Data Science & Business Informatics.
Intelligent systems for pattern recognition course (760AA)

Abstract

In this project we review our path towards building a convolutional network under rigid hardware constraints. We provide a review of the research papers that influenced our module choice and eventually propose a novel hybrid version the AL-NET architecture [8] with the well-established MobileNETV3 [11] encoder. Our model combines the efficient bottleneck modules of MobileNetV3 with a lightweight asymmetric decoder and skip connections fusion scheme of AL-NET. The result is a model capable of accurately producing a segmentation map of the same size as the input image with only 310 thousands parameters, which is approximately the same number of parameter of MobileNetV3 and around 90% less parameters than AL-NET. We compare our architecture to the standard segmentation architecture of MobileNetV3 and to the standard U-Net model [19], the most popular model among Kaggle practitioners for this dataset. On our test set, our model achieves a Dice score of 85% outperforming standard MobileNetV3 (84% Dice) and U-NET (82% Dice).

1 Introduction

Brain tumors are the second greatest cause of death according to the World Health Organization. It is a source of mortality for which diagnosis and treatment require extensive sophisticated diagnostic by expert raters and expensive medical technology. Magnetic Resonance Imaging (MRIs) provide a detailed image of multiple slices of one's brain and are commonly used by doctors to diagnose brain tumor area. However, manual segmentation is time consuming and suffers from significant inter-rater variance. The goal of automatic brain tumor segmentation is to obtain precise segmentation from the MRI images. The benefits of obtaining precise and reliable measurements about the tumor range from statistical and predictive tasks to treatment planning and tumor monitoring. Automatic and non-stochastic segmentation of brain tumors are thus needed.
From the machine learning point of view, accurately segmenting brain tumor is a challenging task given that their border are often fuzzy and hard to distinguish from healthy

tissues. Tumors can appear anywhere in the brain, have any shape, form or contrast. The application of convolutional network for segmentation task is well studied by the community. Since Havei et al showed the potential of fully convolutional networks for brain tumor segmentation in 2015 [9], more than 110 thousands publications have been indexed on semanticscholar.org. Outside the scope of tumor segmentation, research and publications related to innovative convolutional network architectures and modules has grown exponentially in the past decade. Many papers pushed the boundaries of performance reachable with deep convolutional networks; however this gain in performance usually came at a heavy cost in term of computational requirements. More recently, many approaches try to maintain high level of accuracy whilst drastically reducing computational requirements, mainly focusing on reducing the number of parameters and multiplication & addition operations (Mult.Adds).

The dataset object of this project was gathered by [2] in which they used a U-Net model to extract brain tumor segmentation mask and investigate statistical relationships between the shape of the tumor and a patient genomic data. Along with the dataset, the authors provide the implementation code and pre-trained weights for the model. Furthermore, on the Kaggle page of the dataset, more than two hundreds code notebooks with deep learning models are already implemented on this dataset. From a sample of notebooks we analysed, almost all were standard U-Net implementation [19].
Throughout the making of this project, our objective and method repeatedly changed as we had to adapt to our hardware limitations. With only our laptop's CPU and sporadic access to Google Colaboratory free GPU, we quickly realized we could not run, in reasonable time, a classical machine learning experimental project consisting in designing, training and performing rigorous model selection on multiple high performing state-of-the-art models. After investigating potential strategies, we decided on a more theoretical approach to model selection by researching innovative lightweight modules and architectures in the deep learning literature and applying them to the task at hand. We hereafter provide a review of the research papers that influenced our module choice and eventually implement a MobileNetV3 model which we then modify to adapt it to our task. The final model we obtain is a hybrid of the encoder-decoder architecture of AL-Net [8] with the well-established MobileNETV3 + LR-ASPP [11] as encoder.

2 Research

Our research was primarily based on papers available on the platform semanticscholar.org. We used the citation counter of the platform to measure how much a paper was innovative and influential. Given the popularity of U-net, we started by trying to understand the evolution of innovations that led to the U-Net structure and the developments that followed it. The amount of literature we found quickly grew intractable and we hereafter only consider a sample of the papers we found most influential for our task.

Whilst the popularity of convolutional networks takes off around 2012 with the groundbreaking results of AlexNet [16] for image classification, the popularity of brain tumor segmentation takes off slightly later, around 2014/15. Some of the early segmentation models proposed such as Havaei et al.[9] and Pereira et al.[18] were in practice classification models as they took an input patch around each pixel and output a classification for that single pixel. Havaei et al. used the fully convolutional network which did not use dense layers on the top and a dual branch architecture with one branch focused on low-level feature maps to learn about local details and the other focused on obtaining high-level feature maps to extract contextual information.

Shelhamer et al. [20] introduce image-to-image learning where the model outputs a full segmentation map, significantly speeding training time. The authors achieved it by using upsampling layer where the high-level feature maps are bilinearly upsampled to obtain pixel-wise output. Shelhamer et al. also introduce skip connections from coarse low dimensional feature maps to upsampled feature maps to help the model recover localized information. Shelhamer et al. work was designed to exploit pre-trained heavyweight models recently developed such as VGG [21] and Inception [22] and adapt them to perform pixel-level segmentation. This strategy made sense at the time considering that these models offered state-of-the-art performance but were extremely expensive to train.

Building upon the work of Shelhamer et al., Ronnenberger et al.[19] introduce the U-Net architecture. U-Net is a symmetric encoder-decoder designed specifically for medical image segmentation. The contracting path consists of repeated 3x3 convolutions with pooling operations, with the number of filters doubling every time the feature map is halved by the pooling. The expanding path is the same but pooling operators are replaced by convolutional transpose operators. Convolutional transpose are different from upsampling. Whilst upsampling operators simply repeat the rows and columns by a certain factor without any learned parameter, convolutional transpose are the opposite of convolutions: they have trainable weights and learn to map 1 input pixel to kxk output pixels. U-Net also expands the skip connections to concatenate every decoder input with the output of its respective encoder. Such a structure is particularly effective in medical imaging such as brain segmentation as skip connections at every level help reduce the loss of low-level feature information (eg. for detailed boundary segmentation) whilst enabling learning translational invariant global contextual information (eg. for recognizing the types of tumor) through the repeated pooling and low-dimensional convolutions.

Since its publication in 2015, the U-Net architecture has been incredibly influential. With more than 60 thousands papers indexed with the term ”U-Net” in their title and over 25 thousands only in the last three years, U-Net is still a go-to baseline model for image segmentation. The development based on U-Net were countless. We found papers proposing U-Net variations with every innovative modules developed for convolutional models such as Inception, Res-Net [10], Dense-Net [15] and others. A new series of ”hybrid” models combining multiple types of convolutional modules both at encoder and decoder level. Some solutions we initially considered for our project were to improve U-Net's boundary awareness with multi-branch or dual loss or both such as[1] or expand the representational capabilities of the model by exploiting cross-channel information of the input such as 3D medical images such as [28, 17].

The main characterization of U-net, and the one that remained almost intact in all its variation, is the encoder-decoder style architecture. We were thus motivated to chose an encoder-decoder architecture for our project. We did however also explore potential alternative to the encoder-decoder architecture. One of its main antagonist in the image segmentation literature is the DeepLab model by Chen et al. [3]. Whilst U-Net and most other network have the same encoder of image classification models, DeepLab advocates for minimizing as much as possible the need for downsampling and upsampling operations by using a module specifically designed for image segmentation: the multi-scale context aggregation module (Yu et al. [24]).

The context module uses dilated convolutions to aggregate multi-scale contextual information without losing resolution. This is possible because dilated convolution allow exponential expansion of the receptive field without down-sampling the feature map. Image classification modules like VGG learn multi scale contextual information by repeated pooling operations. Whilst downsampling operations are desirable in an image classification task, dense prediction task calls for multi-contextual reasoning together with full-resolution output. Yu et al. propose a context module that is a theoretical rectangular prism of convolutional layers with no pooling or sub-sampling. Thus, in theory the feature maps would have the same input size along all the architecture and their would be no loss of resources on encoding and decoding. The trick relies on compensating the lost multi-scale contextual information (previously obtained by the pooling operators) by using dilated convolutions that effectively enlarge the receptive field. Most importantly, dilated convolutions increase the receptive field of the convolutions without increasing the number of parameters. The receptive field of a dilated convolution of size n x n with dilation rate r is calculated as ((n - 1) * r + 1)$\hat{2}$. For example a 3x3 dilated convolution with dilation rate 3 has a receptive field of 36. As a side note, we noticed that Yu et al. theory connects with Goodfellow et al. (Deep Learning, page 336) who argue that pooling could be considered as a infinitely strong prior probability distribution. Like any prior, it is not useful if its assumption is inaccurate. Our task relies on preserving precise spatial information therefore pooling operators could set a wrong prior.

Coming back to DeepLab, it has a VGG style encoder and it uses dilated convolutions in two ways. First in introduces a module that learn multi-scale contextual information by applying dilated convolutions at different rate through multiple parallel branches called Atrous Spatial Pyramid Pooling (ASPP). This module is applied at the end of the encoder as a way to further extract rich contextual information. Secondly, it uses dilated convolutions on the last two VGG block on the encoder to avoid the last two downsampling steps. Thus the encoder outputs feature maps at 1/8 of the original image instead

of 1/32 as it would have been the case with standard VGG encoder.
Whilst the result looked promising, the DeepLab implementation was far away from the rectangular prism architecture without any downsampling envisaged by Yu et al. By admission of the same Yu et al. in a following paper [25], "Operating at full resolution throughout, with no downsampling at all, is beyond the capabilities of current hardware." which we felt was counter intuitive at first given that the dilated convolutions has the same number of parameters as regular convolutions; however, as the authors say, when feature map resolution is increased by a factor of 2 in each dimension, the memory consumption of that feature map increases by a factor of 4, thus limiting to large scale usage of dilated convolutions across all the network. A further indication of the limitations of dilated convolutions is that in their following papers (DeepLabv2 and DeepLabV3 [4, 5]), Chen et al. shift their DeepLab model to a proper encoder-decoder architecture with skip-connections. The only part of DeepLabv1 remaining is the ASPP module.

The above part motivated our choice to proceed with an encoder-decoder architecture. Training U-Net with its over 7 million parameter was highly unpractical on our hardware. Furthermore, many implementation and pre-trained U-net were available for our dataset, making the task not particularly compelling. We thus decided to investigate the literature regarding state-of-the-art lightweight encoder-decoder convolutional models. Like we did previously, we first researched the most influential papers, then thoroughly investigated its components and the reasons behind their implementation. For the sake of brevity we will only present a small selection of the theory that motivated our choice of the modules we later implemented.
The literature concerning lightweight models mainly takes two path. Some try to compress the model whilst others try to make existing modules more efficient and thus require less parameters. The two main ways to render model more efficient we investigated are (1) factorization of the convolution operation, (2) reducing redundancy.
Concerning the former, we start by Chollet's Xception model [6], where the author showed that depthwise separable convolution could effectively approximate standard convolution. Whilst a standard convolution applies a three dimensional convolution across all channels at the same time. A depthwise separable convolution is a factorization of a convolution in two parts. First, a depthwise convolution applies a single convolution for each single input channel at a time. Then multiple pointwise convolution combine the previously computed feature maps into the required channel size. The assumption behind Chollet's proposal is that the mapping of cross-channel correlations and spatial correlations in the feature maps can be entirely decoupled. Indeed a flat convolution performed on a single channel feature map is actually learning spatial correlation; accordingly, a pointwise convolution on a multi-channel feature map is learning a cross-channel correlation. Chollet showed that this assumption makes can help the model's representation capabilities whilst significantly making it more efficient. Mathematically, the gain in terms of reduced

number of parameter is significant: considering an input feature map DxDxM convolved with N kernels of size KxK. A standard convolution would have K*K*M*N parameters. A separable depthwise convolution has K*K*M parameters for the depthwise convolution and M*N parameters for the pointwise convolutions, totaling M*(K*K+N) parameters. Doing the ratio and simplifying the equation we get that the gain in reduce parameters is (K+N) / (K*N).
Mainly based on depthwise separable convolution is the MobileNet [12] architecture. The MobileNet class of families are purposely designed to be extremely lightweight. MobileNetv1 introduces a ResNet style image classification model specifically designed to use depthwise separable convolution and shows that such models achieved competitive performance. MobileNetV2 adds inverted residual with linear bottleneck. The inverted residual is a simple yet effective solution to reduce memory consumption by connecting subsequent modules on the thin feature map instead that on the feature map with many channels. The linear bottleneck instead refers to the insertion of a linear bottleneck layer in the module. The bottleneck block is now composed of a 1x1 expansion convolution which creates a high dimensional activation space, followed by the depthwise convolution and a 1x1 projection layer that acts as a bottleneck. The authors show that, assuming the data manifold we are interested in lies in low dimensional space, the most effective way to capture it is by using a high dimensional activation space followed by a bottleneck layer with linear activation. MobileNetV3 builds on the former. It adds a "squeeze-and-excite" [13] module inside the bottleneck module. Squeeze-and-excite (SE) is a particular architectural unit that focuses on capturing more effectively channel-wise relationship. It adds parameters to each channel of a convolutional block so that the network can learn weights for each feature map. SE takes an input feature map, it squeeze each channel to a single scalar with global pooling, it uses a first fully connected layer with a Relu non-linear activation, it then uses a second fully connected layer with a sigmoid activation to provide each channel a gating function. The result of this side network is then fused back to the the original feature maps it was first applied to. This enables the model to adaptively re-calibrate channel-wise feature response. SE is a powerful tool that can be easily be fitted to a pre-existing model with almost no additional cost. In MobileNetV3's blocks, we are interested in modeling channel-wise feature response after the initial expansion convolutions, before the 1x1 projection layer. A SE unit is thus fitted on the output of the depthwise convolution bottleneck block. MobileNetV3 also replaces Relu non-linear activation with a non-linear function called the swish. It also introduces a new lightweight decoder based on the DeepLab ASPP previously mentioned.

Concerning the reduction of redundancy, we investigated the potential usage of densely connected module in our implementation. We were particularly interested by the low redundancy architecture of DenseNet along with its "collective knowledge" philosophy. We investigate it further and analysed various lightweight architecture derived from from it. However, after playing around with both the MobileNetV3 and DenseNet/CondenseNet architectures, we eventually we decided to proceed with the MobileNetV3 architecture. Our choice was primarily motivated by (1) the amount of research and testing that went into the development of MobileNetv1 and its refinements MobileNetv2&v3: in the MobileNet papers, its authors clearly motivate each single architectural and hyper-parameter choice with empirical data. Furthermore MobileNetv3 is results of multiple repeated successive papers offering continuous refinements of the same architecture which, by our research, is not common in other state-of-the-art lightweight models. (2) The fact that the implementation are natively implemented in Tensorflow yielding wide availability and support for customizing the model with Keras class API. (3) MobileNetV3 provides a design for a lightweight segmentation head tailored for MobileNetV3.

3 Data Pre-processing

The data concerns the segmentation of glioma, which is the most common type of brain tumor. Gliomas are classified with grades according to the severity of the tumor. Low grade gliomas (LGG) grow slowly and have low probability of spreading. The most common type of brain scans to analyse LGG are MRIs which are composed of multiple pulse sequences: for every scan of a slice of brain we obtain multiple images at different contrasts. In our dataset, we have three sequences per scan: pre-contrast sequence, FLAIR (fluid attenuated inversion recovery) and post-contrast sequence. LGG have low contrast leak and we can only distinguish between normal and abnormal tissue. This explain the binary mask provided in the training data as ground truth. We have between 10 to 80 scans for each of the 110 patients. Every scan has three channels, one for each pulse sequence, the middle channel being FLAIR. The ground truth segmentation mask provided is associated to the FLAIR sequence. The pre-contrast and post-contrast sequence are added as channel side channels to provide extra information.

For pre-processing the data we followed the same steps as the authors of the dataset (Buda et al. [2]). The first step was skull-stripping skull stripping. This is done to let the model focus only on the relevant data: the soft tissue within the cranial cavity. Secondly they used a convex hull post processing to get smoother segmentation of the extracted brains. Buda et al. provide a separate pre-trained U-Net on their Github repository to perform skull stripping on the data. We thus used a U-Net architecture and loaded their weights to perform inference on our brain scans to remove the skull. Given we had no test for evaluation we had to rely on visually checking the results. The sample we initially checked made us believe that the skull-stripping model did perform well; however, when we were later moved on training our segmentation model, we could not understand why our model was unable to learn. After checking many hypothesis, we found that the problem was in the data. Indeed the skull-stripping U-Net was actually removing the tumor part in some scans along with the skull. Indeed some tumors have

very similar contrast to the skull and thus the model got confused. Even applying the convex hull did not solve the issue as significant parts of tumor were still missing. We visually inspected hundreds of images and estimated that about 20% of the segmented brain were missing a part of tumor. Given we had no time or resources to retrain a U-Net for skull-stripping, we decide to "repair" the biased segmentation masks. We used the fact that brains have an ellipse shaped structure. We fitted a minimum ellipse to contain the entire segmentation map produced by the skill-stripping model. We then merged the fitted ellipse with the segmentation map by applying a convex hull. We finally filled newly created shape and used it as segmentation mask of the skull-free brain scan. In Figure A.1, we show the difference between raw input, the original biased output from the pre-trained U-Net and our recovery of the entire brain structure.
Finally we oversampled brain scans with tumor in training data. This was done both for re-balancing the number of brain scans with tumor versus without tumor and for preventing overfitting which our model was suffering from. Oversample was done by applying both random rotation and scale variation at a rate of two synthetic images for each tumorous brain scan.

4 Model implementation and results

We designed the MobileNetV3 architecture with code from its official Keras Github repository. We used the segmentation head proposed by the authors of MobileNetv3 which is an extremely lightweight version of the ASPP module described earlier they called Lite Reduced ASPP (LR-ASPP). Our initial implementation followed the exact design choices found in MobileNetv3's paper for the MobileNetv3_small setup. As suggested by the authors for a segmentation task, we removed the last encoding blocks to reduce downsampling by a factor of two. The encoder consists in an initial set of 16 filters with a 2x2 stride followed by nine bottleneck blocks. Three of the latter perform convolutions with a 2x2 stride. The output of the encoder is thus a feature map at 1/16 of the original input size. The output is fed to the LR-ASPP module. LR-ASPP has two branches each extracting contextual information at different scales. The fist branch keeps the feature maps at the same size and performs 128 1x1 convolutions. The second branch reduces the feature map to a size of 1x1 and uses 128 1x1 convolutions with sigmoid activation. The two branches are then merged and upsampled. The final step before outputting the segmentation map consist in adding the output of the two branches with the skip connection from the 1/8 feature map from the encoder.

We did not attempt any modification to parameters of the architecture. This was due in part to our limited access to a GPU but primarily because we did not think there was a significant need for it. The authors of MobileNetV3 provide exhaustive theoretical and empirical evidence motivating their architectural choice and every decision is a trade-off between computational cost and accuracy. For instance, the number of filters in the first layer was tuned to 16 by the authors in order to keep a large enough initial filters bank for edge detection. The dual branch of LR-ASPP have 128 filters each which expand the encoded feature map into high dimensional feature space at different scales. Such a high number of filters is vital to obtain rich semantic features maps.

We did however have to modify the decoder given that the original LR-ASPP only outputs image at 1/8 of the original input dimension. We performed model selection to evaluate different potential solutions. We split the data into train (60%), validation (30%) and test (10%). We used the Dice similarity coefficient as performance metric, it is an F1 score widely used for segmentation task. We used the negative Dice coefficient as loss function, again widely used and differentiable.

The two main solution we compared are (1) keeping the current architecture and bilinearly upsample the last feature map by a factor of 8 and (2) implement a sightly more articulated skip-connection architecture and directly upsample by 16. Both solutions implied to further upsampling the output feature map to reach the original input size. Whilst we consider the first solution for its simplicity, we designed the second solution with the theoretical background from U-Net. Indeed the MobileNetv3 architecture for segmentation was not designed specifically for medical image segmentation where we require precise boundary information. Such low level information can be mostly found in the early layers of the encoder. Looking at the MobileNetV3+LR-ASPP scheme (Figure A.3) we see that it has only a single skip connection. We hypothesized that the model for our task could benefit from having a slightly more complex skip connection architecture enabling it to retrieve more low-level information. We used the skip conntectoin fusion architecture of AL-Net (Figure A.4) which seemed a reasonable compromise between the hyper simplicity of MobileNetv3's decoder and the multiple skip connection concatenations of U-Net that inevitably increase computational cost.

For both solutions we performed hyperparameter tuning by hand as Gridsearch would have been unfeasible. With the same hyperparameter setting, the second solution obtained a higher performance both on the validation set and the test set. We can not say if the difference in validation Dice is statistically significant but our modified model with extra connections performed marginally better on the model assessment on the test set. We report the results of validation and test performance in Table 4 along with the results obtained with the pre-trained U-Net from Buda et al. on our dataset. We also report the test accuracy the authors reported in their paper [2]. We reported the latter for fairness considering that the pre-trained U-Net we used was trained on data that was pre-processed in a different way, thus the low Dice coefficient could be misleading. Still, comparing when comparing our results, we achieve a significant difference of more than 3 percentage points between our top model and U-Net's Dice score claimed by Buda et al. The difference is even more impressive when considering the colossal difference in in number of trainable parameter between the two models: 7 million for U-Net versus the 312 thousands for our modified MobileNetv3. We provide a sample of the first predictions

2*Models	2*# of trainable parameters	Hyperparpameters					DICE coef	
		opt	lr	mom	nest	bn	DICE VAL	**DICE TEST**
MobileNetv3 + LR-ASPP	310 thousands	SGD	0.09	0.95	Yes	Yes	83.49%	**84.76%**
MobileNetv3 + LR-ASPP with extra skip connections	312 thousands	SGD	0.09	0.95	Yes	Yes	84.01%	**85.26%**
U-Net (pre-trained)	7 milion	-	-	-	-	-	71.12%	**74.12%**
U-Net (claimed)	7 milion	-	-	-	-	-	-	**82%**

Table 1: Final segmentation results. opt: optimizer, lr: learning rate, mom: momentum, nest: Nesterov momentum, bn: batch-normalization after each convolutional layer. U-Net claimed refers to Dice score reported by Buda et al. in [2]

on the test in Figure A.5.

5 Conclusion

Based on the research and results obtained through out this project, we conclude that (1) the research regarding lightweight models has made incredible progress since Ronnenberger et al. introduced U-Net. U-Net should not be considered a default go-to for segmentation tasks but rather a theoretical pillar of encoder-decoder architectures. Investing time in researching and applying modern lightweight architecture can be a winning strategy, especially in case of constrained resources. (2) There is a need for general-purpose lightweight convolutional architectures to use as baseline models, especially among student who do not have access to premium online GPUs. We chose the MobileNet family not because it was the best performing or most innovative model, rather because it had a clear documentation and had a relatively simple structure, which allowed for an easy implementation and high flexibility to modify it as we needed. Increasing the number of such well-established lightweight models would certainly be beneficial for the community and allow future students to perform more rigorous experimental research on convolutional neural network by comparing different models. (3) MobileNetv3 proved it-self to be a good candidate for a lightweight general-purpose model. It is simple yet flexible and extremely efficient. We were able to modify it to best fit our task and obtained highly competitive results.

References

[1] Ahmed Mahmoud Gab Allah, Amany Mahmoud Sarhan, and Nada Mohamed Elshennawy. Edge u-net: Brain tumor segmentation using mri based on deep u-net model with boundary information. *Expert Systems with Applications*, 2022.

[2] Mateusz Buda, Ashirbani Saha, and Maciej A. Mazurowski. Association of genomic subtypes of lower-grade gliomas with shape features automatically extracted by a deep learning algorithm. *Computers in biology and medicine*, 109:218–225, 2019.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin P. Murphy, and Alan Loddon Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:834–848, 2016.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *ArXiv*, abs/1706.05587, 2017.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision*, 2018.

[6] François Chollet. Xception: Deep learning with depthwise separable convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2016.

[7] Dan C. Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, 2012.

[8] Xiaogang Du, Yinyin Nie, Fuhai Wang, Tao Lei, Song Wang, and Xuejun Zhang. Al-net: Asymmetric lightweight network for medical image segmentation. In *Frontiers in Signal Processing*, 2022.

[9] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron C. Courville, Yoshua Bengio, Christopher Joseph Pal, Pierre-Marc Jodoin, and H. Larochelle. Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, 35:18–31, 2015.

[10] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[11] Andrew G. Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324, 2019.

[12] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *ArXiv*, abs/1704.04861, 2017.

[13] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:2011–2023, 2017.

[14] Gao Huang, Shichen Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Condensenet: An efficient densenet using learned group convolutions. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2752–2761, 2017.

[15] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2016.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

[17] Hengxin Liu, Guoqiang Huo, Qiang Li, Xin Guan, and Ming-Lang Tseng. Multiscale lightweight 3d segmentation algorithm with attention mechanism: Brain tumor image segmentation. *Expert Syst. Appl.*, 214:119166, 2022.

[18] Sérgio Pereira, Adriano Pinto, Victor Alves, and Carlos Alberto Silva. Brain tumor segmentation using convolutional neural networks in mri images. *IEEE Transactions on Medical Imaging*, 35:1240–1251, 2016.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015.

[20] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2014.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[22] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, D. Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2014.

[23] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *CoRR*, abs/1511.07122, 2015.

[24] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *CoRR*, abs/1511.07122, 2015.

[25] Fisher Yu, Vladlen Koltun, and Thomas A. Funkhouser. Dilated residual networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 636–644, 2017.

[26] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2017.

[27] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support : 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, S...*, 11045:3–11, 2018.

[28] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2016.

Appendix A

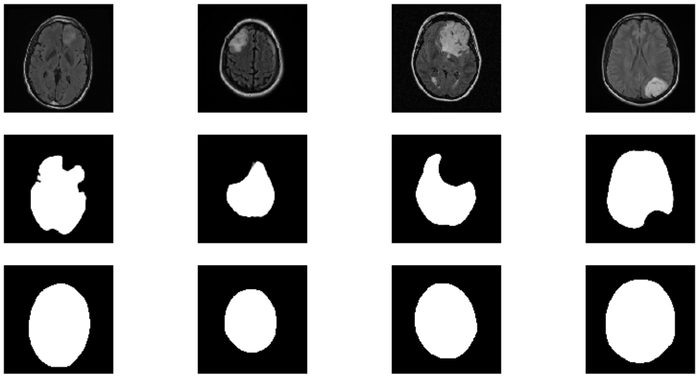

Figure 1: Proprocessing steps. First row shows raw input images; second raw shows the output of the skull-stripping pre-trained U-Net of Buda et al.[2]; third row shows the result of our information recovery algorithm.

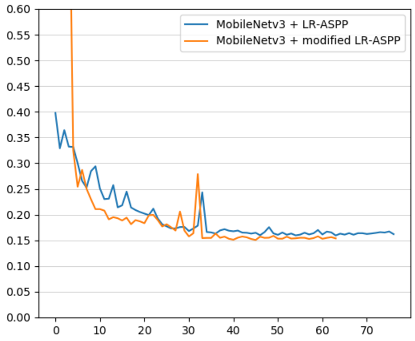

Figure 2: Learning curve showing validation Dice performance for the two solutions we compared.

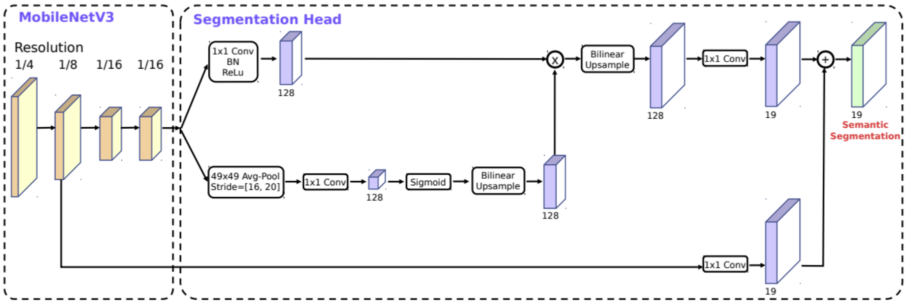

Figure 3: MobileNetV3 with LR-ASPP decoder scheme. Source [11]

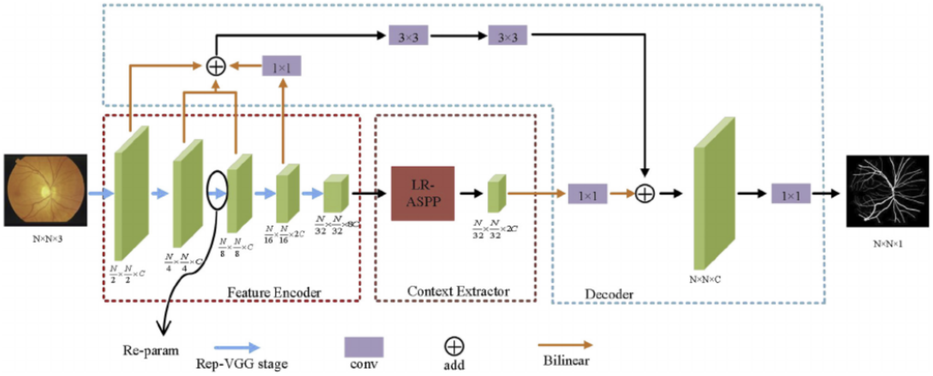
3×3
3×3
1×1
N×N×3
$\frac{N}{2}\times\frac{N}{2}\times C$
$\frac{N}{4}\times\frac{N}{4}\times C$
$\frac{N}{8}\times\frac{N}{8}\times C$
$\frac{N}{16}\times\frac{N}{16}\times 2C$
$\frac{N}{32}\times\frac{N}{32}\times 8C$
Feature Encoder
LR-ASPP
$\frac{N}{32}\times\frac{N}{32}\times 2C$
Context Extractor
1×1
Decoder
N×N×C
1×1
N×N×1
Re-param
Rep-VGG stage
conv
add
Bilinear

Figure 4: Al-Net scheme. Source [8]

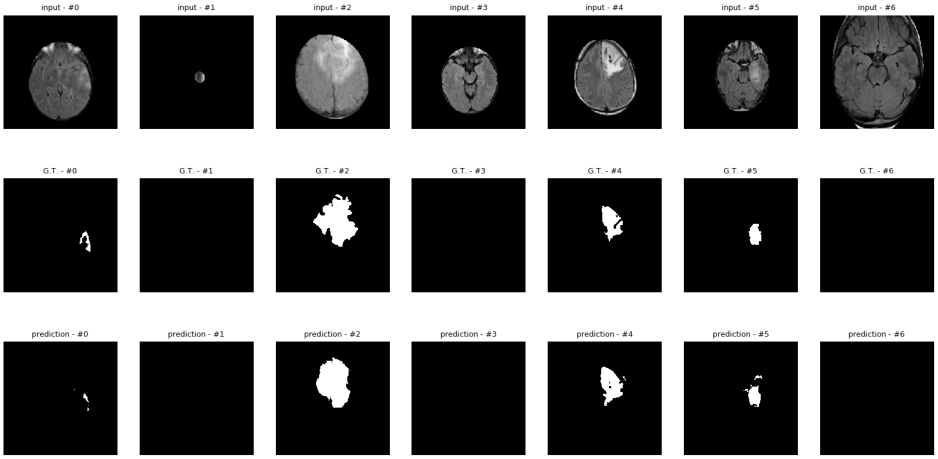

Figure 5: Sample of predictions of our top performing madel on the test set.