Ultrasound Nerve Segmentation : A kaggle competition Final Project Report

Khaoula Belahsen - Nadir El Manouzi
khaoula.belahsen@ensta-paris.fr - nadir.elmanouzi@ensta-paris.fr
Master MVA - 2020
Ecole Normale Superieure Paris-Saclay
Course: DLMI - Deep Learning for Medical Imaging

I. Introduction

We participated in an open challenge on Kaggle called "Ultrasound Nerve Segmentation" (https://www.kaggle.com/c/ultrasound-nerve-segmentation/overview) which took place in 2016 and allow us to submit our algorithms and compare them with the ones of other participants. The goal is to build a model which can identify nerve structures in a dataset of ultrasound images of the neck. Accurately identifying nerve structures in ultrasound images is a critical step in effectively inserting a patients pain management catheter. The main related works for the problem are variants of U-Net. In this project, we began with the U-Net architecture and then we focused on recent state-of-the-art models : R2U-Net [2], U-Net++ [3] and Attention Residual U-Net [4]. In order to improve the performance of each model, we tuned the hyperparameters and developed an augmented dataset by doing transformations like flips and rotations. As we wanted to have more information about the quality of each model, we evaluated the performance not only with the dice coefficient but also with the sensitivity and the Jaccard similarity.

II. Problem Definition

The goal of our work is to engage in the kaggle competition "Ultrasound Nerve Segmentation" by tuning and fitting the best performing model in a task of image segmentation.

Starting from hand-annotated masks on pixelised ultrasound images, the goal is to predict these masks using the best model possible.

The loss function used for almost all our models is the negative of Dice coefficient. The leaderboard score is the mean of the Dice coefficients for each image in the test set. The formula is given by :

$$s = \frac{2|X \cap Y|}{|X| + |Y|}.$$

where X is the predicted set of pixels and Y is the ground truth.

In order to enrich our study of the models, we also explored the use of other metrics such as Recall (to account for images without masks) and IoU (Jaccard similarity) which is very similar and actually positively correlated with the dice coefficient.

III. Related Work

In our project, we studied several models and their architectures. We will briefly describe their corresponding papers in this section. More details will be given in the next section.

The first model studied in this project is U-Net [1] as it is one of the most popular deep learning technique for medical image segmentation. This model provides great results for this task while working not necessarily with big datasets. The architecture is U-shaped (see Fig. 5). The first path is a contracting network and it captures context. The second path is symmetric as an expanding network

and it is used in the purpose of localization. For this goal, the features of the contracting path are combined with the upsampled output.

A more recent model, R2U-Net [2], leverages the previous architecture with recurrent residual convolutional networks. In object recognition tasks, Recurrent CNN and variants have shown superior performance. For the segmentation task, R2U-net has demonstrated better performance on segmentation tasks than U-Net using different benchmarks. Some advantages of this model are that residual units help building a more efficient deeper model and also that the feature accumulation method helps extract better feature representation in particular low-level features.

IV. Methodology

The first step we took in this project was to find a recent end-to-end implementation of a submission with a good result on the leaderboard. We chose the one in this link : https://www.kaggle.com/gbatchkala/edward-tyantov-edited-py/. This is the implementation of a U-Net variant with 2 heads (one for predicting the probability of nerve) and the score obtained by the author is 0.67765 on the private leaderboard. We chose to run all our code on Google Colaboratory and after adapted the previous code we got a score of 0.66921. We took this model as a baseline and the goal during this project was to find a model with a better score. For this purpose, we have used implementations of the papers described in the previous section, adapted them to our dataset, searched the best hyperparameters and used data augmentation.

A. Data collection

A code loads the images and saves them as NumPy array files. The dataset augmented after various transformations is also saved as npy files.

B. Data augmentation

We introduced different functions to perform data augmentation on our datasets. Especially spatial transformations including : flip (vertical, horizontal, random) and rotations. We didn't include any augmentation on the RGB and colouring, the black and white being very disctinctive of the masks.

Therefore, we created 6 data augmentation functions : random_rotate, vertical_flip, random_flip, random_rotate90, random_horizontal_flip, vertical_flip_cv2.

Below are some examples of data augmentation :

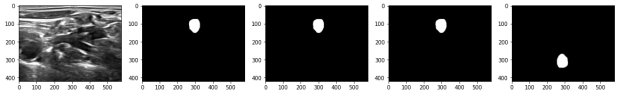

Fig. 1. Results of data augmentation : from left to right ground truth, ground truth mask, random rotation, random horizontal flip, random vertical flip.

C. U-Net

Before feeding the images to the U-Net network, we used a script to resize all images to a 96 x 96 shape. The network architecture we used with this dataset is $32 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 32$. Using the Adam optimizer with a learning rate of 1e-5 and a batch size of 32 allowed us to get a score of **0.58607**. The code runned for 46 epochs (around 16 seconds per epoch). This score is almost the same as the ones people working on this challenge got with the U-Net architecture. In the next paragraph, we describe a variant of U-Net which allowed us to have a file for predicting the probability of nerve presence in an image. Using it gave a much better score with **0.64789**. In order to use it, one needs to first run the next model to generate this file and then run the submission script with two_outputs = True instead of False.

D. U-Net inception with 2 heads

A variant of U-Net with inception blocks and an auxiliary head used for predicting the probability of nerve presence allowed us to get the highest score during this project. The link to the implementation we used is shared in the previous section. Because the dataset contains images where the nerve is not present (almost half of them), having a network predicting the nerve presence and using it at inference time as a filter to predict an empty mask or not is a great idea and leads to

better convergence. The mask presence branch is in the middle of the U-Net, after the encoding unit.

Because of the two heads structure, the final loss is the weighted sum of the negative dice coefficient loss and the binary cross entropy loss. The binary cross entropy being weighted two times less. Finally, all images were resized to a 80 x 112 shape.

We obtained a score of **0.68589** on the leaderboard using the Adam optimizer with a learning rate of 0.0045 and a batch size of 64. We studied the effect of removing the possibility at inference time to predict an empty mask when the probability of nerve presence was lesser than 0.5. We observed a significant loss with a score of only 0.58930.

Then, we tried using the augmented dataset in order to have a better score with this model. Using the same optimizer (Adam) with the tuning of the batch size (32, 64 and 128) and the learning rate (1e-5, 1e-4) only achieved a score of 0.61850. We turned to the RMSProp optimizer and the results were better. With the tuning of the batch size, the learning rate and the parameter rho (lr = 1e-2, batch size = 64, rho = 0.9) we got a similar score to the model which was trained on the original dataset with **0.68311** but we didn't achieved to show a better performance.

E. R2U-Net

The first main difference between R2U-Net and U-net is that in the R2U-Net recurrent convolutional layers are used instead of forward convolutional layers in both the encoding and decoding paths. The second one is the choice of only using concatenation operations and removing cropping and copying units. The architecture of this model is showed in Fig. 6. The unit used in the R2U-Net model is shown in Fig. 2 (d).

For the implementation part, we used an available code from a github repository (link shared in the notebook). Like with the U-Net architecture, all images were resized to a 96 x 96 shape. As the number of parameters was too big (96,056,577), we reduced it by changing the initial number of features from 64 to 16. The total number of parameters was then reduced to 6,003,009. We made a test with the learning rate (2e-4) for the Adam optimizer and the binary cross entropy loss which are used in the reference paper for lung

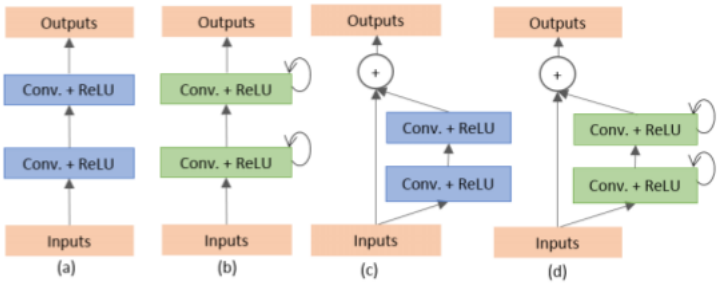

Fig. 2. Different variant of convolutional and recurrent convolutional units (a) Forward convolutional units, (b) Recurrent convolutional block (c) Residual convolutional unit, and (d) Recurrent Residual convolutional units (RRCU). Source : [2]

segmentation. We got with this configuration only 0.46913 on the leaderboard and decided to turn to other configurations. We used batch normalization, changed the batch size (128, 64, 32) and tried different learning rate of the Adam optimizer (1e-6, 1e-5). Finally, we achieved a score of **0.51214** with a batch size of 64 and a learning rate of 1e-5. The model runned for 15 epochs with around 50 seconds for each epoch. Using the probability of nerve presence file generated by the previous model, a much better score was achieved with **0.64770**.

F. Unet++ or Nested Unet

One other model that we used for performance comparison is Unet++. It uses the Dense block ideas from DenseNet to improve U-Net.

UNet++ differs from the original U-Net in three ways [3]:

- It has convolution layers on skip pathways, which enables to link the feature maps between encoder and decoder (shown in green in fig 7)
- It improves the gradient flow due to the presence of dense skip connections on skip pathways (shown in blue in fig 7)
- It is implemented with a deep supervision possibility. This enables model pruning and is intended to improve the performance compared to using only one loss layer (shown in red in fig 7)

Skip connections used in U-Net connect the feature maps between encoder and decoder directly, which can lead to uniting sementically

dissimilar feature maps. However, with UNet++, the output from the previous convolution layer of the same dense block is fusioned with the corresponding up-sampled output of the lower dense block. This makes optimization easier.

Here again, we used the dice coefficient as a metric and loss functions for our unet++ model.

To optimize this model, trained over 50 epochs, we performed hyperparameters tuning on the following parameters :

- Batch size : 32 and 64
- Learning rate : 1e-5, 1e-4, 2e-4, 3e-4
- Optimizer : Adam, RMSProp

We obtained the best results for on our validation set with Adam optimizer with a learning rate of 2e-4. The loss function was Dice coefficient. The model completed training with 9,041,610 trainable parameters in 18 minutes; each epoch took approximately 25 seconds. We obtained a score of **0.63241** on the leaderboard.

G. Attention Residual Unet (AttResUnet)

Attention ResUnet can be seen as a combination of different previously developped models :

- **Unet model** : previously described in this report
- **Attention Gates** : Need to pay attention by Jetley et al. introduced the concept of an end-to-end-trainable attention module. Attention gates are used in natural image analysis and NLP. Attention is used to perform class-specific pooling. It can amplify the relevant regions, thus demonstrating superior generalisation and better classification performance.
- **Residual block** : incorporated in this version of Attention-Unet.

To improve segmentation performance, Attention Unet relies on object localization models to separate localization and subsequent segmentation steps. This can be achieved by integrating attention gates on top of U-Net architecture, without training additional models.

As a result, attention Unet can improve model sensitivity and accuracy without significant computation overhead. The model can supress feature responses in irrelevant background regions.

V. Evaluation

The dataset used in this project is composed of a large training set of images where the nerve has been manually annotated by humans. This dataset can be downloaded directly from the Kaggle webpage. In the training set folder, there are 5635 images (.tif) with their corresponding binary mask showing the Brachial Plexus (BP) segmentations if the BP is present. A csv file gives the training image masks in run-length encoded format. The test set is made up of 5508 images.

Some of the other metrics used in image segmentation are the sensitivity and the Jaccard similarity. We decided to use them in order to compare our models on the validation set.
The operation used to compute the recall (=sensitivity) is :

$$SE = \frac{TP}{TP+FN}$$

where TP is the number of true positives i.e. the number of times the model predict correctly the positive class and FN is the number of false negatives i.e. the number of times the model predict incorrectly the negative class. In our keras implementation, in order to compute this metric on the training and validation set, we added *tf.keras.metrics.Recall* to the list of metrics.
The Jaccard similarity is computed as :

$$JS = \frac{|GT \cap SR|}{|GT \cup SR|}$$

where GT refers to the ground truth and SR to the segmentation result. To use this metric, we added *tf.keras.metrics.MeanIoU(num_classes=2)* to the list of metrics.

A. Quantitative results

The summary of our results on the original dataset can be found in Table I and the ones on the augmented dataset are shown in Table II. The best result we got was obtained using only the original dataset with a score of 0.68589. The model used is the variant of the U-Net model with two heads and inception blocks. For reference, this score allows us to be ranked 60 over more than 800 participants. This model has less parameters and achieved the highest score with a large margin. After all the hyperparameter tuning done on state of the art

TABLE I

MODEL PERFORMANCE WITH THE ORIGINAL DATASET

Models	Nb of parameters	With the probability of nerve presence file	Sensitivity (validation set)	Jaccard similarity (validation set)	Dice coefficient (validation set)	Dice coefficient (leaderboard)
U-Net	7,759,521	No	0.6125	0.5833	0.6062	**0.58607**
-	-	Yes	-	-	-	**0.64789**
U-Net inception 2 heads	1,865,676	No	0.6078	0.7076	0.5986	**0.58930**
-	-	Yes	-	-	-	**0.68589**
R2U-Net	6,003,009	No	0.6373	0.5988	0.5896	**0.51214**
-	-	Yes	-	-	-	**0.64770**
U-Net++	9,041,601	No	0.6778	0.6635	0.5109	**0.63241**
AttResNet	9,786,857	No	0.4776	0.6690	0.1045	**0.55389**

TABLE II

MODEL PERFORMANCE WITH THE AUGMENTED DATASET

Models	Sensitivity (validation set)	Jaccard similarity (validation set)	Dice coefficient (validation set)	Dice coefficient (leaderboard)
U-Net inception 2 heads	0.0666	0.5185	0.0942	**0.68311**
U-Net++	0.0634	0.5181	0.0957	**0.62423**

models, we weren't able to reach this score. And the dataset obtained after augmentation didn't help us in this objective. One can also note the effect on the other models of the file used to predict the nerve presence generated with this U-Net 2 heads model. With this file, more than 6 points are obtained on U-Net and more than 13 points on R2U-Net.

B. Qualitative results

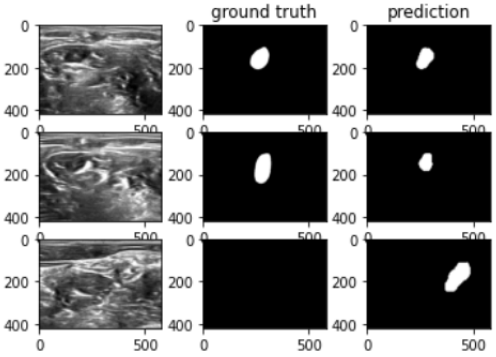

Fig. 3. Images with ground truth masks and predictions from the U-Net with 2 heads model. The dice coefficients are 0.88 for the first image, 0.59 for the second image and 0.0 for the third image

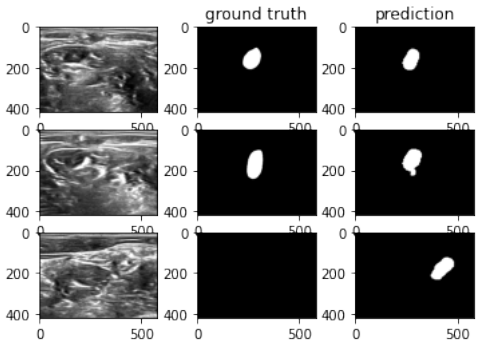

Fig. 4. Images with ground truth masks and predictions from the U-Net++ model. The dice coefficients are 0.89 for the first image, 0.78 for the second image and 0.0 for the third image

For the qualitative assesment of two of our models (U-Net 2 heads in Fig. 3 and U-Net++ in Fig. 4), we extracted three images and their respective masks from the training set and showed the prediction masks generated.

The segmentation results for the first image are good for the two models with a dice coefficient of 0.88 and 0.89. For the second image, the U-Net 2 heads has some difficulties to segment completely the nerve (score of 0.59), one can observe that only half of the nerve is segmented. With a score of 0.78

for the same image, the U-Net++ performs much better with a larger mask predicted.
The two models fails with the last image predicting the nerve presence with a mask located in the same area whereas in the ground truth there is no nerve segmentation at all. The dice coefficient is therefore 0.0 for the two models on this image.

VI. Discussion

In this project, we studied different state-of-the-art models for image segmentation in the context of a kaggle competition starting from the classic U-Net to more advanced model like Attention Residual U-Net. The highest score was obtained with a variant of U-Net with two heads and inception blocks. We weren't able to reach its performance with the other models and the data augmentation process. In future, other data augmentation variations can be done to further improve performance. And as we focused on hyperparameter tuning to have better scores, we could explore some architecture transformations of the state-of-the-art models studied in this work.

VII. Figures

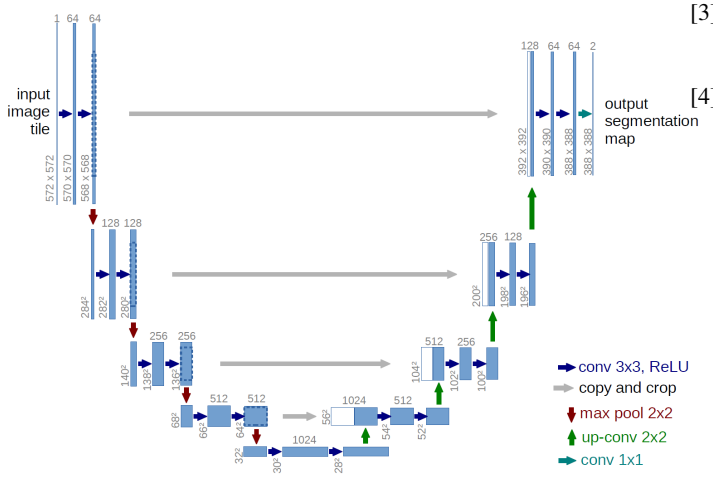

Fig. 5. U-Net architecture

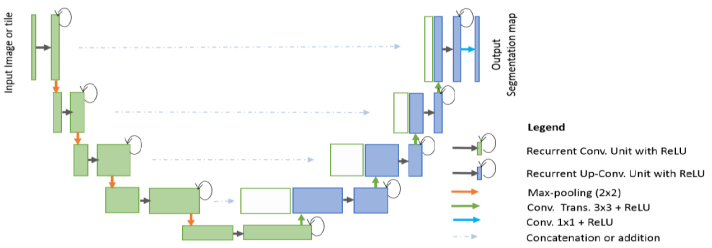

Fig. 6. R2U-Net architecture

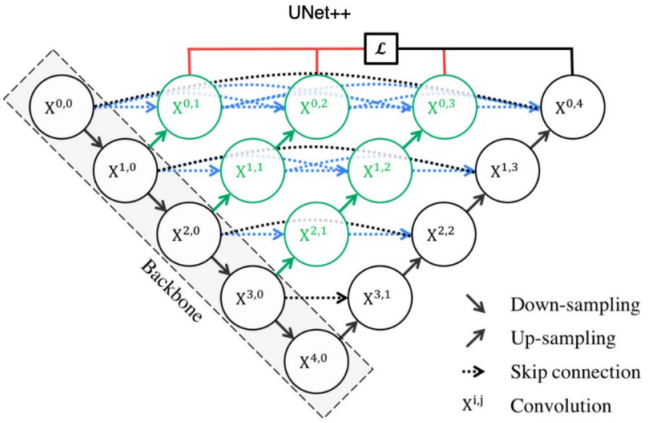

Fig. 7. U-net++ architecture

References

[1] Olaf Ronneberger, Philipp Fischer and Thomas Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, CoRR, 2015

[2] Md. Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M. Taha and Vijayan K. Asari, Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation, CoRR, 2018

[3] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, Jianming Liang, UNet++: A Nested U-Net Architecture for Medical Image Segmentation, 2018

[4] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, Daniel Rueckert, Attention U-Net: Learning Where to Look for the Pancreas, 2018