Application of Deep Learning Techniques for Prostate Magnetic Resonance Imaging (MRI) Segmentation

Khaled Ahmed
Department of Computer Science and Engineering
Innopolis University
Innopolis, Russian Federation
k.mohamed@innopolis.university

***Abstract*—This paper documents the results of a series of experiments applying Computer Vision (CV) and Deep Learning (DL) on a number of public patient data sets aiming to achieve better diagnostic reliability for cancer.**

***Index Terms*—unet, deeplearning, cancer, prostate, brain**

I. Introduction

The progress of Artificial Intelligence opened several opportunities to solve complicated tasks quickly and autonomously. This is not the exception for medicine where the correct usage of collected data can help to find a patient diagnosis and give the required help in time. This project concentrates on the task of segmentation using MRI data of the prostate. Segmentation allows us not only to classify the image into the prostate cancer diagnosis but also to predict the location and shape across the image. We use Deep Learning, a part of machine learning (and subset of AI) that uses Neural Network models to pre-process, extract the features, and segment an input in an end-to-end style.

II. Related Works

Most of the work done utilises convolutional neural networks (CNN) that are pretrained for prostate segmentation, after passing it through a U-Net. We adopted a similar approach to previous works [7] of splitting 3-T MR scanner outputs into lower dimensional images.

Many public datasets explicitly annotate lesions for MRI images of patients (annonimized). The classification in those is done manually with the help of radiologists, and is aimed at providing a better baseline for machine learning assisted approaches to the diagnosis, thus, all of them are confirmed cases.

III. Dataset

We adopt the prostate MRI data from Medical Segmentation Decathlon [1] which contains the standard samples and their masks. More information can be found in Table 1.

TABLE I
Dataset Summary

Dataset Content	Prostate MRI
Dataset Description	Prostate transitional zone and peripheral zone segmentation
Tensor Image Size	4D
# of Images	32

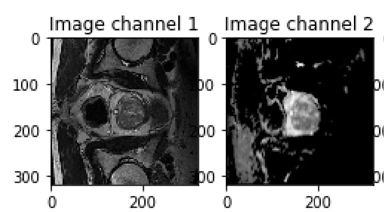

Fig. 1. MRI sample at different channels.

IV. Methodology

In image segmentation, we go through each pixel of the sample and assign it a particular class. As a result, we get the location and the shape of the target mask. This process requires the use of large-scale models because complex problems have to be solved with complex architectural models and a large amount of data. Since we are finding the form and relative location of the target class, models consisting of convolution operations should be utilised.

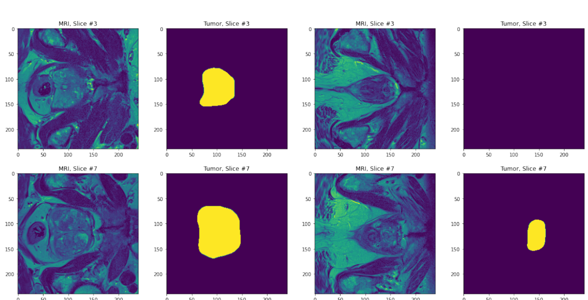

Fig. 2. MRI samples and Tumour masks.

Convolution layers can fit an image of any size and learn the representation pattern of the dataset which is exactly what we want to do. Fine Tuning the pre-trained large models can also be an option since they don't require much computational power.

After investigating existing models, our primary choice is to implement the U-net [2] model. The architecture consists of two parts. Down-sampling is a sequence of convolutions, non-linear functions, and max-pooling operations that reduce the size of the image without trying to lose much information. Up-sampling does the opposite, it brings the image into a higher resolution again which results in a segmentation map. Figure 3 shows the details of the network.

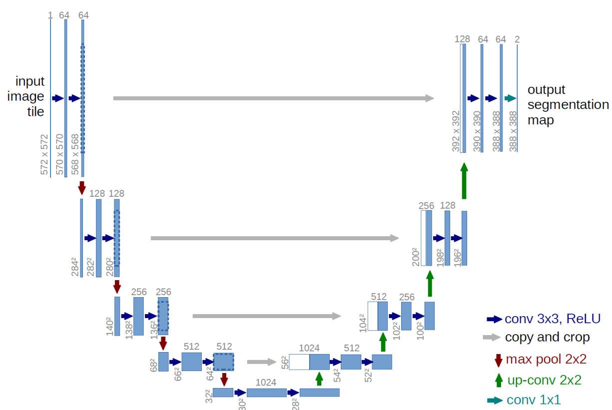

Fig. 3. U-net architecture (example for 32x32 pixels in the lowest resolution). Blue components correspond to a multi-channel feature map. The number of channels is denoted on top. The x-y-size is provided at the lower left edge. White components represent copied feature maps. The arrows denote the different operations.

We implement two versions of the U-net architecture that use a different strategy for the up-sampling process. One is the nearest interpolation operation that creates high resolution using the filling process of the neighbour pixels. The second one uses transposed convolutions which try to reverse the convolution operation [3].

We can improve the vanilla U-net by using a different backbone in the architecture rather than a simple set of convolutions. Optimal operations in the encoder part can get better features in the input which in turn will provide more useful information in the decoder. The decision was to use a residual network architecture, which as result produces a ResUnet [4] network. The overall model is called ResUnet. The network shows great overall performance compared to others in different experiments, especially in medical tasks.

Finally, we wanted to see how the Dilated Densenet architecture [5] is able to solve this task. This is not as complex as previous architectures, but it is composed of dilated convolution operations that are promising in terms of solving difficult tasks.

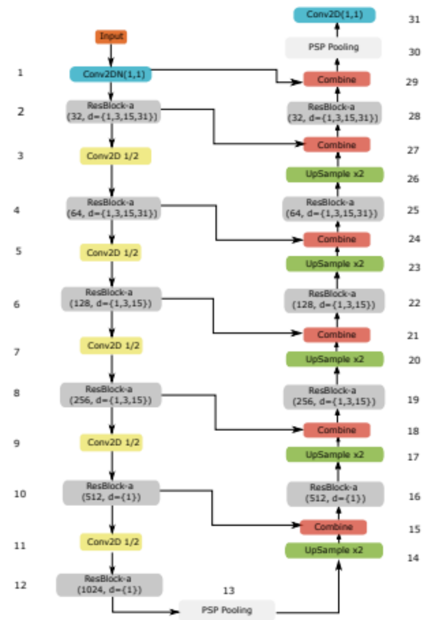

Fig. 4. ResUnet architecture.

To evaluate the models we primarily use dice coefficient metrics [6]. It compares the area between the predicted and target masks. There are again two versions, the one that rounds the areas into the same priority and the soft version that does not apply this rounding. We want to see the behaviour of both during the training and see if the results have high sensitivity. The loss is a custom one and adapted from the dice coefficient using the logarithm operation. Basically, it converts the discrete metrics into a continuous function that is possible to optimise. Finally, to evaluate and see how well segmentation-trained models can diagnose prostate cancer, we use binary classification accuracy. If it predicts the non-zero area on positive diagnosis and zero area on negative, the prediction is correct.

V. Project Repository

Notebooks, training data, images, and report relating to this project is publicly available on Github: https://github.com/Ex-o/mri-segmentation

VI. Experiments

We split the data into train, validation, and test parts by 80%, 10%, and 10%. We apply a basic transformation to the dataset at some probability to increase the variety of data. First of all, we get rid of the 2nd channel as the input and output of the model have to be the same. The transformations are: resizing to (240, 240) dimension, normalisation, cropping, rotation, and flipping. The probability of transformations is set to be high. Finally, the biggest issue we had was dealing with 3D data. The models are designed to handle the images. Also, batches of different size samples cannot be handled while the data has different depths throughout images. Therefore, we decided to slice each sample along the levels and therefore obtain several images for each sample. This solves two problems at once, limited data and the dimensionality issue.

The models were made from scratch by trying both the Tensorflow and Pytorch libraries. Most of the work was done from scratch, but for some of them, we used the ready hub models or publicly available code that is referenced whenever used.

Training is done in a GoogleColab environment that provides limited GPU usage. The GPU is a 16GB Tesla T4. To solve the limitation problem, we add periodic check pointing in order to continue from the last stage. Over fitting is avoided by using the check pointing of the best validation models during the process by looking through the validation metrics. To not train for very long, an early stopping callback is added. Hyper parameters are found via trials and errors. Fixing all, tuning one, and then repeating with the others in a sequential manner.

VII. Analysis and Observations

We first made the approach using PyTorch, but after failing to train the models we the same ideas were easier to implement using Tensorflow. This is an interesting experience, this could be because of incorrect slicing of the dataset in PyTorch, while Tensorflow functions provide an easy path to achieve it.

The change in the metrics during the training is summarised in Figure 5. The Validation score does not drop which suggests a lack of over fitting. The curves are saturated at the end of the training which means that the maximum score is already reached. Regardless of the model, the curves look similar, which in turn conveys the information about how the task is solved by using the gradient descent methods. The problem in the training is that the curves are really unstable and change a lot during the small number of epochs. Also, the validation score seems higher than the training one. The reason is because of the small dataset and the complex models we are training, The gap in data, and network complexity brings this instability.

The first observation that the accuracy of the classification is almost 100% for all the models which suggests the power of the segmentation task and how loose the metrics are for our models. Even though the models fail to learn to predict the perfect shape of the tumor, the ability to find its existence is already a big step forward as it can support the diagnosis. The loss values and dice score are highly correlated (in opposite directions), meaning that the designed loss is a good one.

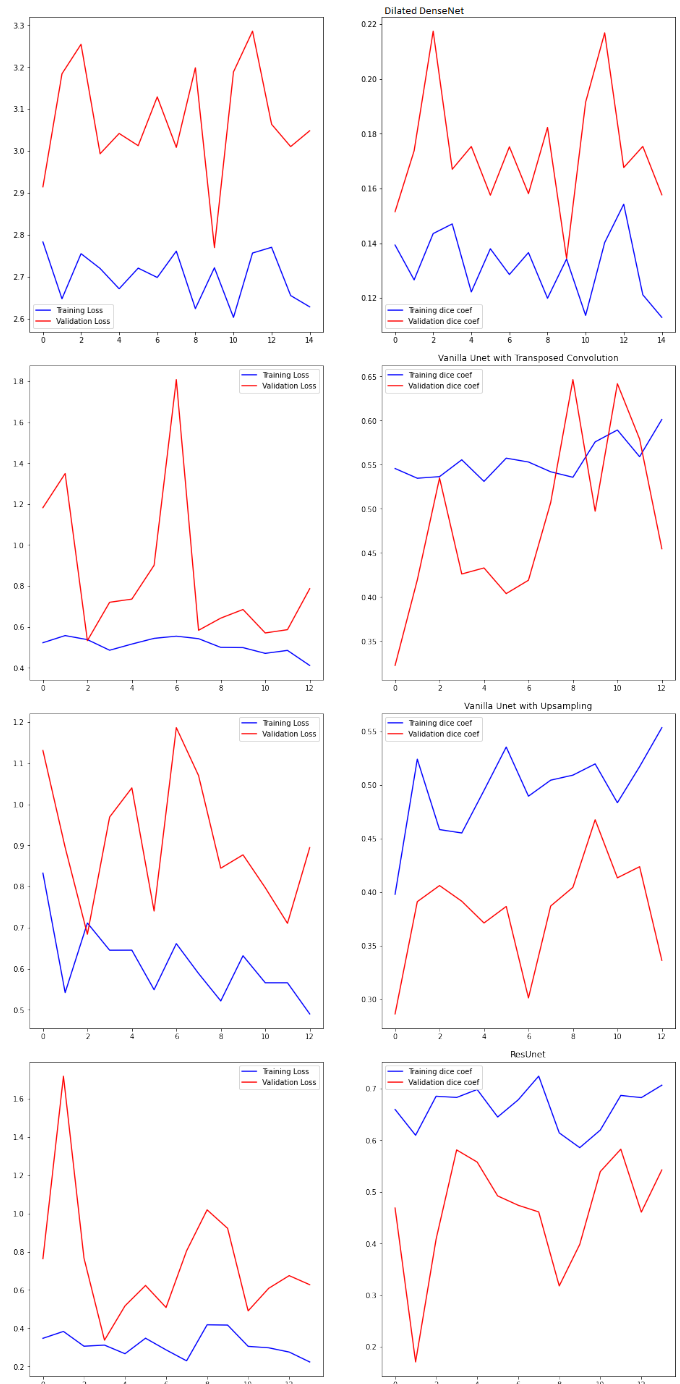

Fig. 5. Loss and dice coefficient curves during the training.

TABLE II
MODEL PERFORMANCE

Model	Loss	Dice Coeff	Soft Dice Coeff	Accuracy
UNet with upsampling	0.3524	0.6034	0.6022	1.0
UNet with Transpose Convolution	0.2954	0.7317	0.6768	1.0
ResUnet	0.8070	0.4580	0.4068	0.97
Dilated Densnet	2.2179	0.0814	0.1230	0.8

The score values are interesting. Dilated Densenet model could not be trained well under several parameters. Even though, it performs well on other tasks, medical diagnosing, the one that has only a single channel sample. The model just could not be trained on the limited and similar data. Although the ResUnet network was very promising, it did not outperform vanilla models. Again, the reason could be the complexity. From the loss curve, we see that it could converge better. For example, the test loss is 0.8 while vanilla models get 0.3 and are not much better in the dice coefficient. This suggests that more training could help with performance. But more training, in turn, lead to over fitting. Both versions of U-Net achieved the highest scores. The scores are very close to each other, meaning that both up-sampling types could do their jobs. But if we become more strict, the transposed convolution one is better. Probably because it is an already implemented function that is trained better on Tensorflow.

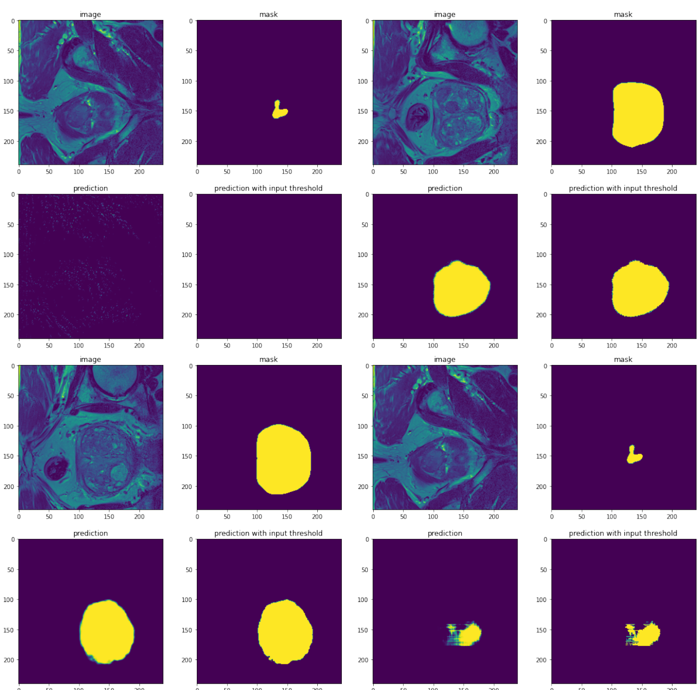

Fig. 6. Model prediction examples.

Figure 6 shows the example predictions of each model on random test samples. The shape and location are precise, but not the exact same. This proves that the task was solved successfully, but there is still a place for future improvement.

VIII. CONCLUSION

In this project, we explored the MRI dataset, analysed the AI task to support the medical diagnosis, and applied 4 deep learning algorithms which could already achieve surprising performance. There were several challenges because of the limited dataset, single channel samples, and 3D data that we had to overcome. We saw how a single network can perform with different architectures and combinations. Also, we compared how two different networks behave under the same environment. This demonstrates how efficient Machine Learning is at classification based diagnosis.

REFERENCES

[1] Antonelli, M., Reinke, A., Bakas, S. et al. The Medical Segmentation Decathlon. Nat Commun 13, 4128 (2022). https://doi.org/10.1038/s41467-022-30695-9

[2] Ronneberger, O., Fischer, P., Brox, T. (2015, May 18). U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv.org. Retrieved November 26, 2022, from https://arxiv.org/abs/1505.04597v1

[3] Transposed Convolution - Dive into Deep Learning 1.0.0-alpha1.post0 documentation. (n.d.). Retrieved November 26, 2022, from https://d2l.ai/chapter_computer-vision/transposed-conv.html

[4] Diakogiannis, F. I., Waldner, F., Caccetta, P., Wu, C. (2020, January 15). ResUNet-A: A deep learning framework for semantic segmentation of remotely sensed data. arXiv.org. Retrieved November 26, 2022, from https://arxiv.org/abs/1904.00592

[5] Sourour Brahimi, Najib Ben Aoun, Chokri Ben Amar, A Benoit, Patrick Lambert. Multiscale Fully Convolutional DenseNet for Semantic Segmentation. WSCG 2018, International Conference on Computer Graphics, Visualization and Computer Vision, May 2018, Pilsen, Czech Republic. ffhal-01786688

[6] Intersection over union (IOU). Hasty.ai. (2022, November 23). Retrieved November 26, 2022, from https://hasty.ai/docs/mp-wiki/metrics/iou-intersection-over-union

[7] Hosseinzadeh, M., Saha, A., Brand, P., Slootweg, I., de Rooij, M., Huisman, H. (2021). Deep learning–assisted prostate cancer detection on bi-parametric MRI: Minimum training data size requirements and effect of prior knowledge. European Radiology, 32(4), 2224–2234. https://doi.org/10.1007/s00330-021-08320-y