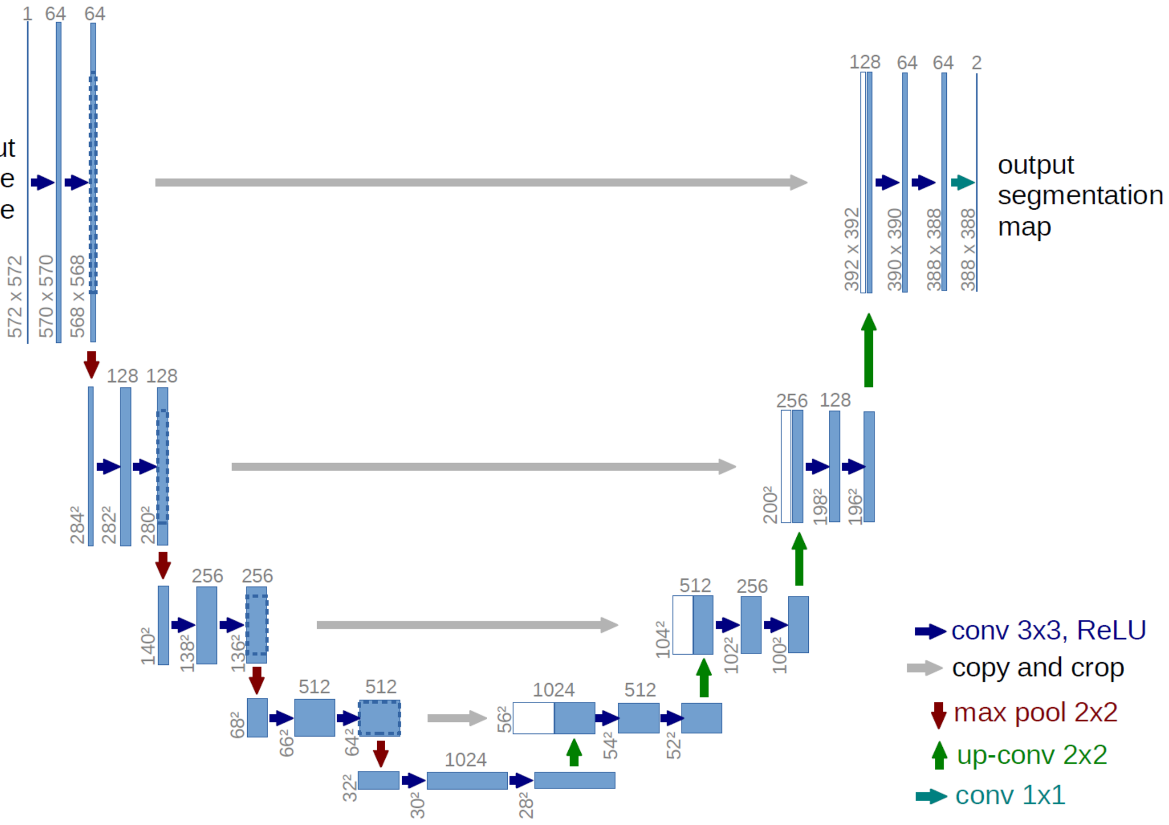
NECST NL1 Kits19 Bio-Imaging Report

“There are more than 400,000 new cases of kidney cancer each year, and surgery is its most common treatment. Due to the wide variety in kidney and kidney tumor morphology, there is currently great interest in how tumor morphology relates to surgical outcomes, as well as in developing advanced surgical planning techniques. Automatic semantic segmentation is a promising tool for these efforts, but morphological heterogeneity makes it a difficult problem.”[1]

The challenge was to segment and find kidney tumors from the scans of over 200 patients. More information can be found on https://kits19.grand-challenge.org/.

Model Architecture

The chosen model architecture was U-Net which is broadly used in biomedical image segmentation and can be thought of an encoder network followed by a decoder network.



[2]

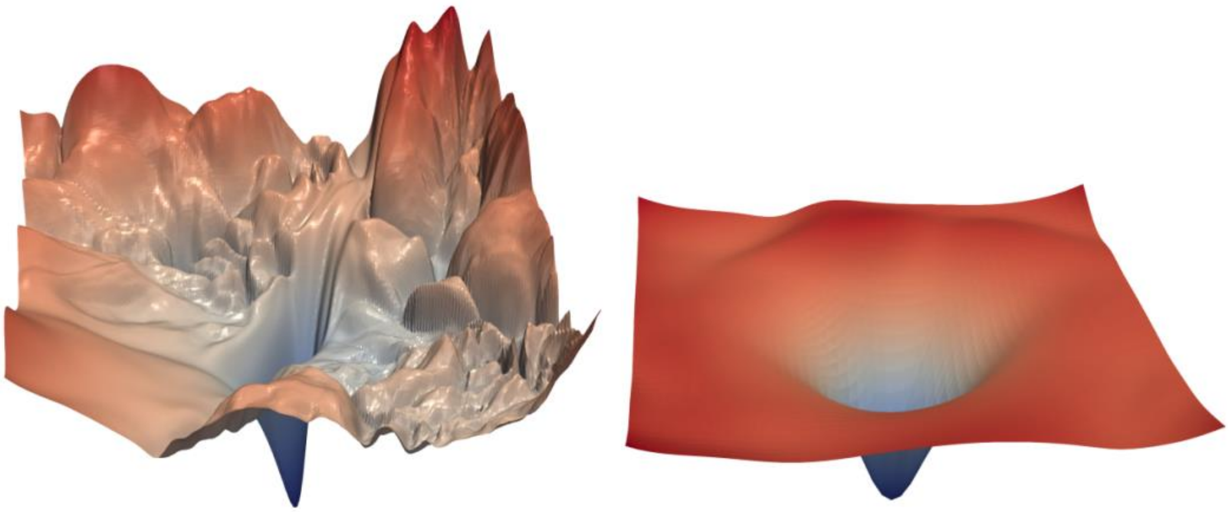
[1] “Kits19 - Grand Challenge.” grand. Accessed December 19, 2022. https://kits19.grand-challenge.org/.

[2] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Lecture Notes in Computer Science, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

As it is shown above, the architecture is purely convolutional with downscaling/upscaling. In the left section, at each block the features double to find and learn the details in the image while on the right side it gets upscaled to get back the original resolution. It is also possible to notice that exactly two convolutional layers are applied in each block.

The architecture also implements skip connections which are fundamental to let the network "remember" the previously learned details during upscaling.

In the project implementation, the cropping phase in the skip connections was not implemented as the images are perfectly divisible by 2 three times (being 512x 512 pixels). This is important because otherwise we would have to round the pixels to the nearest integer and therefore not only implement crops in the left part of the net to make sure that the skip convolutions are of the same size.



(a) without skip connections

(b) with skip connections

[3]

The image shown above represents the loss surfaces of ResNet-56 with and without skip connections skip connections. As it can be seen they make make a huge difference in terms of training stability and therefore they were also added to my implementation.

[3] T, S. (2021) Skip connections: All you need to know about skip connections, Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/08/all-you-need-to-know-about-skip-connections/#:~:text=awesome%20concept%20now.-,What%20are%20Skip%20Connections%3F,different%20problems%20in%20different%20architectures. (Accessed: December 23, 2022).

Hyperparameters

The hyperparameters chosen were:

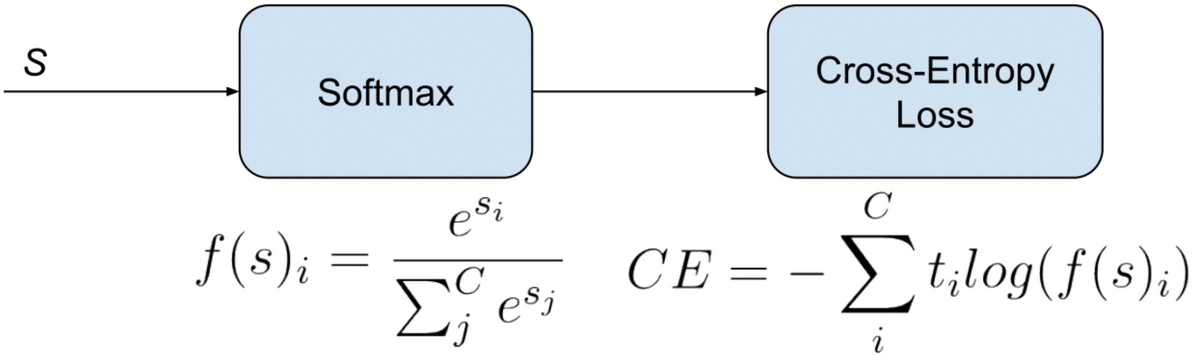
```
learning_rate = 0.002
batch_size = 4
epochs = 4
```

It is important to notice the low batch size due to a lack of memory in the graphics card (computed using cuda) and with any bigger batches resulting in gpu memory being fully occupied therefore causing runtime errors. Batch normalization, in fact, was not used due to the low batch size.

The optimal learning rate was found to be roughly 0.002 after some trial and error in the training of the U-Net. The learning rate is low due to the large amount of images that the net needs to analyze: roughly 500 for each patient.

Loss function

The loss function used in model training was Cross Entropy Loss. Although Pytorch automatically applies Softmax and calculates cross entropy even with multi labelled data (not one hot encoded), the process is the following:



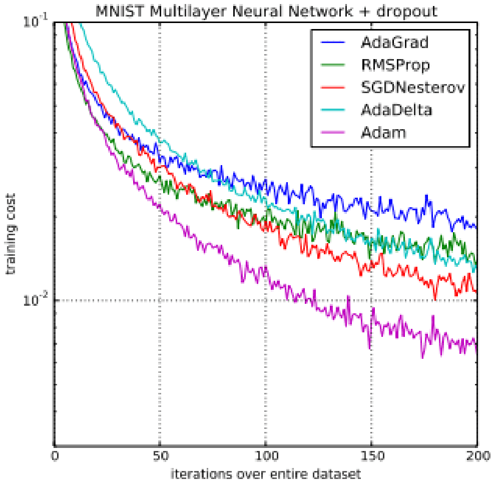
$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$

[4]

Basically, after passing through Softmax (considering one hot encoding), each class is expressed as a probability of a pixel belonging to that class which is then compared to the true value by the Cross Entropy Loss. Based on the difference between these values a penalty score is calculated which is logarithmic in nature: yielding a large score for differences close to 1 and a small score for differences next to 0.

[4] Understanding categorical cross-entropy loss, binary cross-entropy loss, Softmax loss, logistic loss, focal loss and all those confusing names. Available at: https://gombru.github.io/2018/05/23/cross_entropy_loss/ (Accessed: December 23, 2022).

Optimizer



The optimizer used in the project is Adam which combines the advantages of two other versions of stochastic gradient descent:

- Adaptive Gradient Algorithm (AdaGrad).
- Root Means Square Propagation (RMSProp)

Instead of adapting the model learning rates on the mean as in RMSProp, Adam also makes use of the variance. Adam is not only accurate but also greatly reduces training cost as shown on the left.[5]

Code implementation

The code was implemented using Python 3.10 and the latest version of Pytorch with CUDA support 11.7. A section of the U-Net implementation followed from the Pytorch tutorial made by Aladdin Persson[6].

Results and conclusion

Overall, the model performed with good performance measures: even though the model was only trained on the first 50 patients, it achieved a pixel accuracy of more than 98.5% and as high as 99.5% in the following patients

Validation Patient	Pixel accuracy
50	98.5%
51	98.8%
52	99.4%
53	99.2%
54	99.5%

Certainly, the pixel accuracy rating is not the best performance measure and more accurate statistics such as the dice score for each class can be considered as a possible extension to verify the performance

[5] Brownlee, Jason. “Gentle Introduction to the Adam Optimization Algorithm for Deep Learning.” MachineLearningMastery.com, January 12, 2021. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

[6] Persson, Aladdin. “Pytorch Image Segmentation Tutorial with U-NET: Everything from Scratch Baby.” YouTube, YouTube, 2 Feb. 2021, https://www.youtube.com/watch?v=IHq1t7NxS8k&amp;t=1310s.

of the implemented model. The dice score is a measure of similarity of two samples which not only measures the positives you find but also penalizes the false positives found by the model.

Another possible extension would be improving the efficiency of the model, therefore reducing the time taken for training and increasing the number of patients the model trains on up to the full 210 given by the kits challenge.

Bibliography

- "Kits19 - Grand Challenge." grand. Accessed December 19, 2022. https://kits19.grand-challenge.org/.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Lecture Notes in Computer Science, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28
- T, S. (2021) Skip connections: All you need to know about skip connections, Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/08/all-you-need-to-know-about-skip-connections/#:~:text=awesome%20concept%20now.-,What%20are%20Skip%20Connections%3F,different%20problems%20in%20different%20architectures. (Accessed: December 23, 2022).
- Understanding categorical cross-entropy loss, binary cross-entropy loss, Softmax loss, logistic loss, focal loss and all those confusing names. Available at: https://gombru.github.io/2018/05/23/cross_entropy_loss/ (Accessed: December 23, 2022).
- Brownlee, Jason. "Gentle Introduction to the Adam Optimization Algorithm for Deep Learning." MachineLearningMastery.com, January 12, 2021. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.
- Persson, Aladdin. "Pytorch Image Segmentation Tutorial with U-NET: Everything from Scratch Baby." YouTube, YouTube, 2 Feb. 2021, https://www.youtube.com/watch?v=IHq1t7NxS8k&amp;t=1310s.