

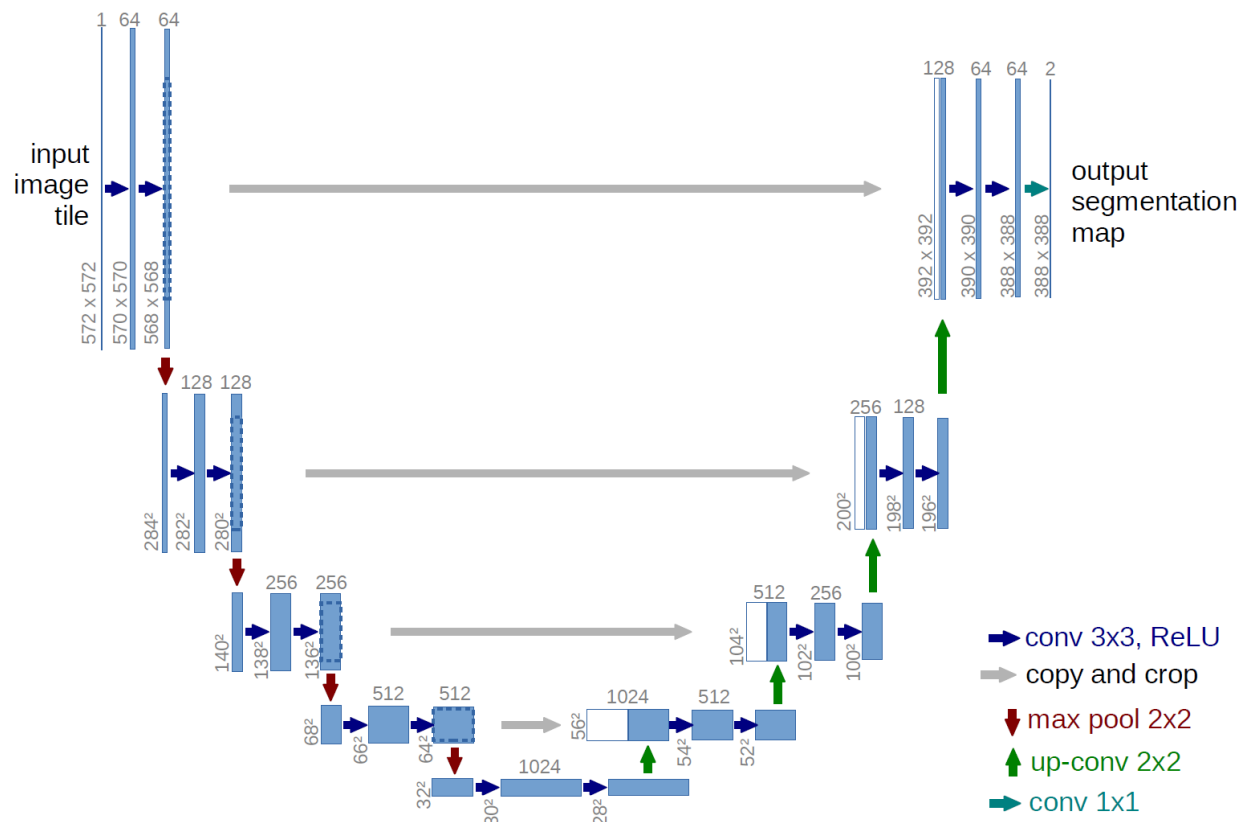
## Report

“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.”<sup>1</sup>

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

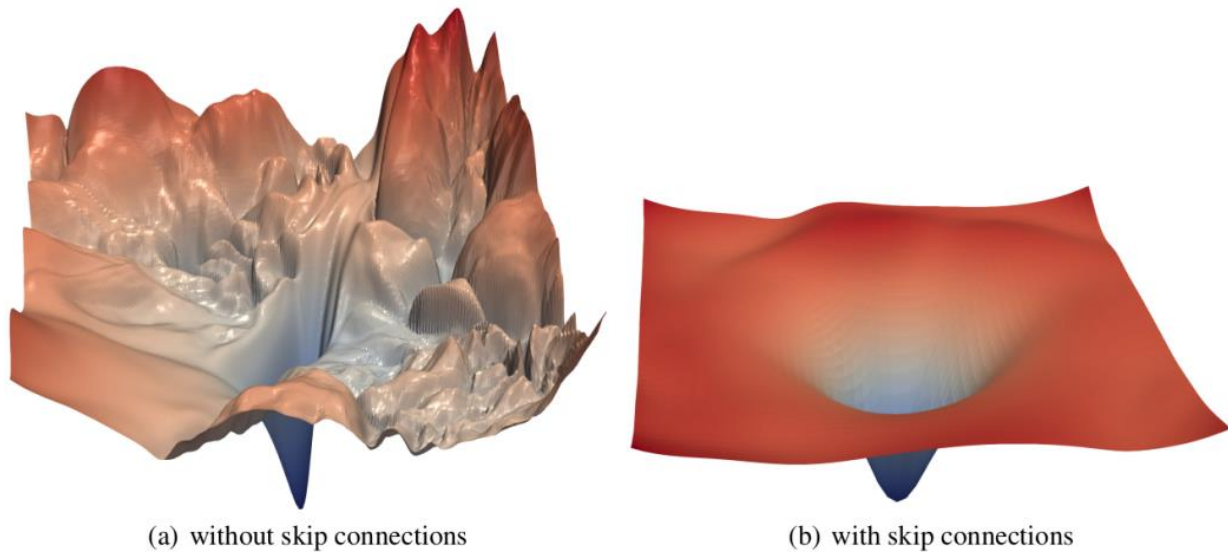
<sup>1</sup> “Kits19 - Grand Challenge.” grand. Accessed December 19, 2022. <https://kits19.grand-challenge.org/>.

<sup>2</sup> 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](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.

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).



3

As shown above, skip connections make a huge difference in terms of training stability and therefore they were also incorporated in my project.

## Hyperparameters

The hyperparameters chosen were:

```
learning_rate = 0.001
batch_size = 4
epochs = 1
```

It is important to notice the low batch size and epochs. The first due to a lack of memory in the graphics card (computed using cuda) with any bigger batches resulting in memory being fully occupied. The

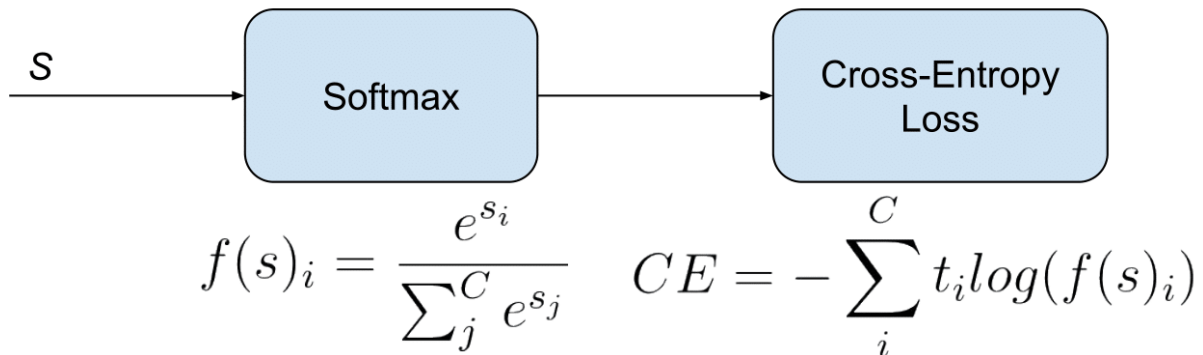
---

<sup>3</sup> 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).

second due to slow training speed, in fact, the training process was executed, in my project, only on the first 20 patients due to a lack of training speed.

## Loss function

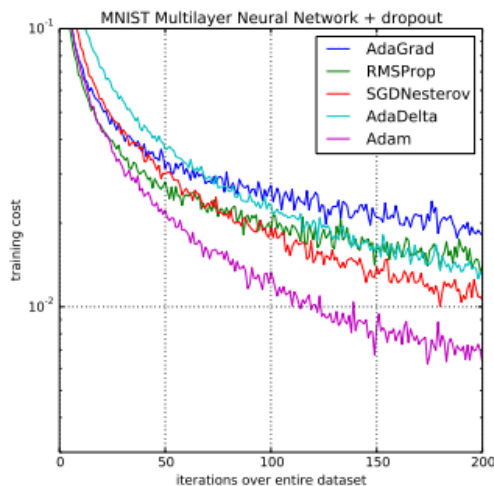
The loss function used in model training was Cross Entropy Loss. Although Pytorch automatically applies Softmax and calculates cross entropy even with multi labelled data (not one hot encoded), the process is the following:



4

Basically, each class is expressed (considering one hot encoding), after passing through Softmax, as a probability of a pixel belonging to that class which is then compared to the true value by the Cross Entropy Loss.

## 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.

<sup>4</sup> 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/](https://gombru.github.io/2018/05/23/cross_entropy_loss/) (Accessed: December 23, 2022).

## Results and conclusion

Overall, the model performed with good performance measures: even though the model was only trained on the first 20 patients, it achieved a pixel accuracy of more than 97% in many instances, for example when checking the predictions for patient 51 it achieved an accuracy rating of: 97.6%

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 of the implemented 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.

---

<sup>5</sup> 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/>.