**Methods**

*Architecture*

U-Net is a segmentation technique originally proposed for medical imaging segmentation in a 2015 paper.(1) The model architecture is straightforward: an encoder, down sampling, and a decoder, for up sampling, with skip connections. As Figure 1 shows, it shapes like the letter U hence the name U-Net. The grey arrows represent the skip connections that concatenate the encoder feature map with the decoder, this assists in the backward flow of gradients and improves training.

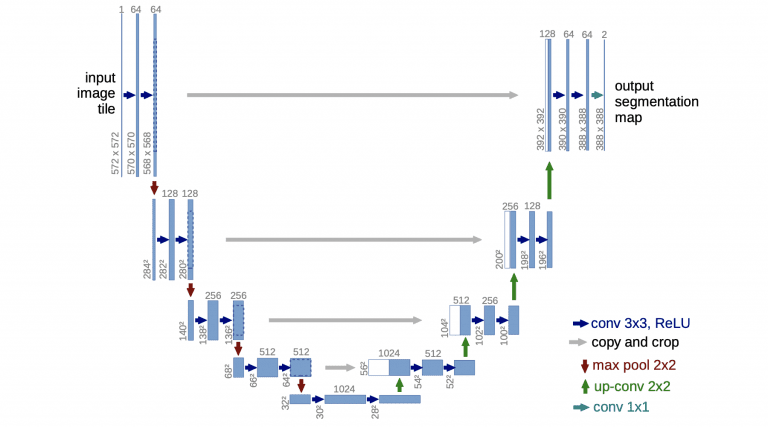


Figure 1. The original U-Net architecture(1)

*Defining a model*

Three iterative blocks will be used in defining the U-Net model, namely the convolution operation block, the encoder block, and the decoder block. With the help of these three blocks, the U-Net architecture can be constructed with ease.

The convolution operation block is used to perform the primary operation of taking the entered input parameters and processing a layer of convolution operations. Arguments, the input for the convolution layer, the number of filters and padding are included. The value of padding as same is used to maintain the same shape of the convolutional layers.

The encoder architecture block will use consecutive inputs starting from the first layer all the way to the bottom. The encoder function is a defined in the convolutional block. Once passed through the convolution blocks, these elements are quickly downsized.

The decoder block will include three arguments, namely the receiving inputs, the input of the skip connection, and the number of filters in the particular building block. The entered input will be up sampled in the model. The receiving input and the newly up sampled layers will be We will then be concatenated to receive the final value of the skip connections. This combined function will then be used to perform a convolutional block operation to proceed to the next layer and return this output value.

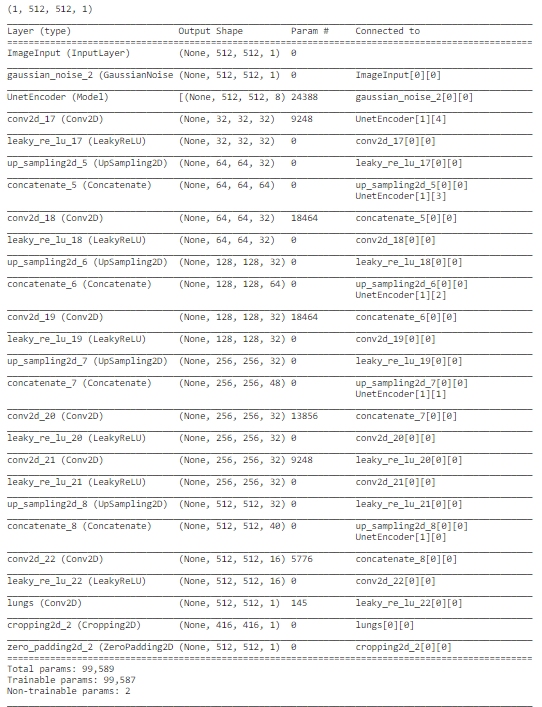


Figure 2. A summary of the layers of U-net used in this paper.

*Training*

Data Split

80% of files from the Montgomery dataset were randomly assigned to training, and the remaining 20% to validation. The trained model is then applied to the all RSNA images.

Augmentation

Data augmentation is the process of modifying, or “augmenting” a dataset with additional data. In this case it is achieved by rotating, blurring, and flipping images in the training dataset. This helps to increase the performance of the model by improved generalising and thereby reducing overfitting. Augmentation can be applied as “pre-processing” before training or as “real time” during training. Other simple augmentation techniques include cropping, sheering, zooming in or out, and adjusting brightness or contrast. In this model, horizontal flip, rotation, shift, shear, and zoom are all utilised.

Loss function

Neural networks may be designed to produce a continuous range of output values while others require a 1/0 output. Mean Squared Error (MSE) loss is ideal for the first type of task, for the second type, a classification task, a different type of loss function would be ideal. Binary Cross Entropy (BCE) loss is suited for a classification task, it penalises wrong but confident outputs in addition to correct but not so confident outputs. BCE is used in this model as image segmentation consists of the classification of each pixel in an image.

Optimiser

The technique for back propagating gradients to update network weights has several options available. Adam was used in this case. It utilises the idea of momentum to reduce the chances of getting stuck in a local minimum. It also uses a separate learning rate for each learnable parameter, which can adapt to how each parameter changes during training. An alternative would be Stochastic Gradient Descent (SGD), it is popular due to its simplicity, and it is lightweight in terms of computational resources however it can get stuck in local minima in the loss function and a single learning rate applies to all parameters.

Controlling the learning process

A hyperparameter is a parameter whose value is used to control the learning process. The learning rate is an example of a hyperparameter. In addition to altering the learning rate with the optimiser, a scheduler can decrease the learning rate if there has not been a reduction in validation loss in set number of epochs. For this model this number, known as the patience, was set to three.

The maximum number of epochs is another hyperparameter that can be altered, in this model it was set to 25. An early stop trigger was set so that if the validation loss has not decreased in 15 epochs the model would terminate.

1. Ronneberger O, Fischer P, Brox T, editors. U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015; 2015 2015//; Cham: Springer International Publishing.