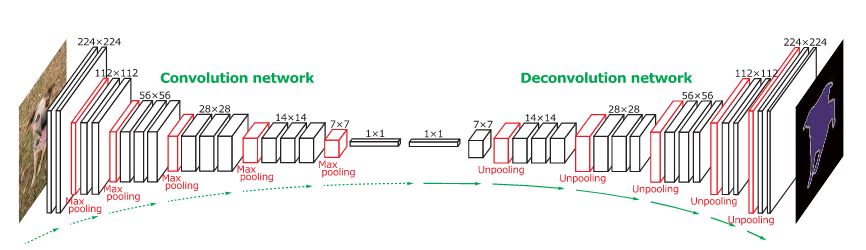
The first part of the proposed architecture was taking the diagnosis samples and segmenting out the Region of Interest which for the given problem is the tumour tissue and after successfully segmenting it out use it for extracting some features that will further be used for detection of a Malignant tumour. For the segmentation, the architecture is inspired from the U-Net model([Paper)](https://arxiv.org/abs/1505.04597). The model used in the proposed system consists of a encoder and a decoder for predicting a pixel level mapping between the diagnosis image and the segmentation mask. For the encoder a Convolution neural network can be used to encode the spatial information and after that a deconvolution neural network will be used to decode the segmentation mask. Multiple architecture were tried for the encoder and decoder network like Alexnet,VGG and ResNet. The decoder network was based on a deconvolution neural network corresponding to encoder architecture like Alexnet,VGG and ResNet.



The segmentation architecture is a Fully Convolutional Network.The original Fully Convolutional Network (FCN) learns a mapping from pixels to pixels, without extracting the region proposals. The FCN network pipeline is an extension of the classical CNN. The main idea is to make the classical CNN take as input arbitrary-sized images. The restriction of CNNs to accept and produce labels only for specific sized inputs comes from the fully-connected layers which are fixed. Contrary to them, FCNs only have convolutional and pooling layers which give them the ability to make predictions on arbitrary-sized inputs.Although the classical FCN used methods like bilinear interpolation and cubic interpolation to convert the bounding box predictions to finely grained pixel level segmentation, the proposed model directly segment out the image on pixel basis with the help of deconvolution network.

The UNet model consists of different modules and operations such as :

**1)Convolution Layer:**

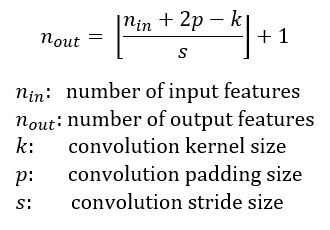
There are two inputs to a convolutional operation

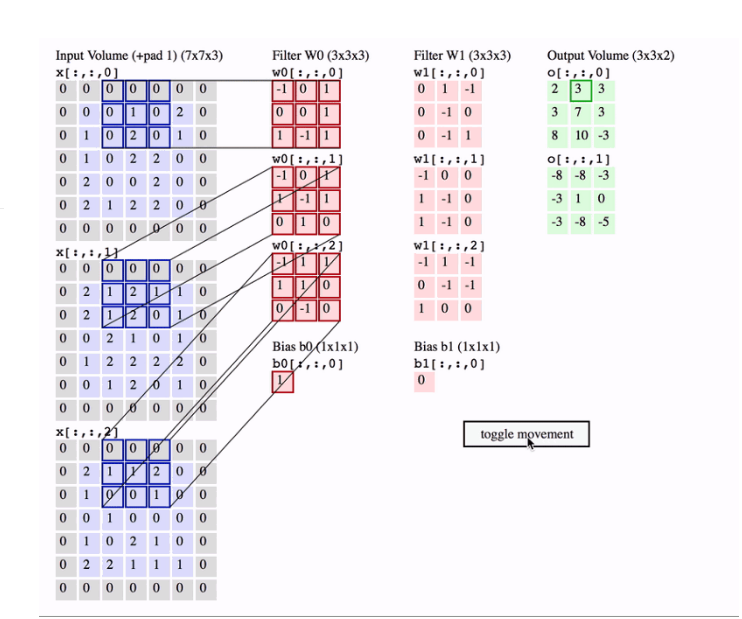
i) A 3D volume (input image) of size (nin x nin x channels)

ii) A set of ‘k’ filters (also called as kernels or feature extractors) each one of size (f x f x channels), where f is typically 3 or 5.

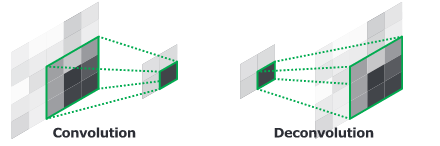
The output of a convolutional operation is also a 3D volume (also called as output image or feature map) of size (nout x nout x k).

The relationship between nin and nout is as follows:





In the above image, we have an input volume of size 7x7x3. Two filters each of size 3x3x3. Padding =0 and Strides = 2. Hence the output volume is 3x3x2. If you are not comfortable with this arithmetic then you need to first revise the concepts of Convolutional Networks before you continue further.One important term used frequently is called as the Receptive field. This is nothing but the region in the input volume that a particular feature extractor (filter) is looking at. In the above GIF, the 3x3 blue region in the input volume that the filter covers at any given instance is the receptive field. This is also sometimes called as the context.To put in very simple terms, receptive field (context) is the area of the input image that the filter covers at any given point of time.



**2)Deconvolution Layer:**

The output of an unpooling layer is an enlarged, yet sparse activation map. The deconvolution layers densify the sparseactivations obtained by unpooling through convolution-like operations with multiple learned filters. However, contrary to convolutional layers, which connect multiple input activations within a filter window to a single activation, de-convolutional layers associate a single input activation with multiple outputs, as illustrated in Figure. The output of the deconvolutional layer is an enlarged and dense activation map. We crop the boundary of the enlarged activation map to keep the size of the output map identical to the one from the preceding unpooling layer.The learned filters in deconvolutional layers correspond to bases to reconstruct shape of an input object. Therefore,similar to the convolution network, a hierarchical structure of deconvolutional layers are used to capture different level of shape details. The filters in lower layers tend to capture overall shape of an object while the class-specific fine-details are encoded in the filters in higher layers. In this way, the network directly takes class-specific shape information into account for semantic segmentation, which is often ignored in other approaches based only on convolutional layers.

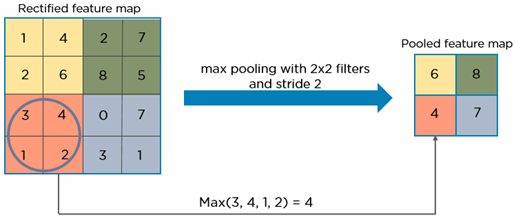
**3)Add Skip Connection(Copy and Concatenate):**

To get better precise locations, at every step of the decoder we use skip connections by concatenating the output of the transposed convolution layers with the feature maps from the Encoder at the same level.  
After every concatenation we again apply two consecutive regular convolutions so that the model can learn to assemble a more precise output

**4)Maxpool Layer:**

In simple words, the function of pooling is to reduce the size of the feature map so that we have fewer parameters in the network.

For example:



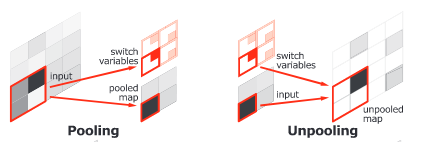
Basically from every 2x2 block of the input feature map, we select the maximum pixel value and thus obtain a pooled feature map. Note that the size of the filter and strides are two important hyper-parameters in the max pooling operation.

The idea is to retain only the important features (max valued pixels) from each region and throw away the information which is not important. By important, I mean that information which best describes the context of the image.

A very important point to note here is that both convolution operation and specially the pooling operation reduce the size of the image. This is called as **down sampling**. In the above example, the size of the image before pooling is 4x4 and after pooling is 2x2. In fact down sampling basically means converting a high resolution image to a low resolution image.

Thus before pooling, the information which was present in a 4x4 image, after pooling, (almost) the same information is now present in a 2x2 image.

Now when we apply the convolution operation again, the filters in the next layer will be able to see larger context, i.e. as we go deeper into the network, the size of the image reduces however the receptive field increases.



**5)Unpooling Layers:**

Pooling in convolution network is designed to filter noisy activations in a lower layer by abstracting activations in a receptive field with a single representative value. Although it helps classification by retaining only robust activations in upper layers, spatial information within a receptive field is lost during pooling, which may be critical for precise localization that is required for semantic segmentation.To resolve such issue, we employ unpooling layers in de-convolution network, which perform the reverse operation of pooling and reconstruct the original size of activations as illustrated in figure. To implement the unpooling opera-tion, we follow the similar approach. It records the locations of maximum activations selected during pooling operation in switch variables, which are employed to place each activation back to its original pooled location. This unpooling strategy is particularly useful to reconstruct the structure of input object.