Student Name: Mohammad Mohammad Beigi
Student ID: 99102189
Subject: Deep Learning

Deep Learning - Dr. E. Fatemizadeh
Assignment 3 -CNN and Vision

Question 1

By assuming zero padding, each convolution operation can be likened to a neuron with 100 inputs and 100 weights. Consequently, each neuron functions akin to a neuron in a dense layer. Thus, if the fully connected network has 5 neurons in its hidden layer, there is no distinction between the networks, assuming learnable convolutions.

In the scenario where the number of neurons in the hidden layer of the fully connected network exceeds 5, it can be asserted that the fully connected network outperforms the convolutional network. Conversely, if the number of neurons in the hidden layer of the fully connected network is less than 5, it can be argued that the convolutional network is superior to the fully connected network. These conclusions are made under the assumption of no overfitting in our networks.

Also note that if convolutions are not learnable, of course fully connected network is better than the other network.

Question 2

a)

$$\text{padding } = 2 \ \& \ 5 \times 5 \text{ conv } \& \text{ stride} = 1 \Rightarrow \text{ Same Convolution}$$

So if the input size is $3 \times 128 \times 128$, output size will be $16 \times 128 \times 128$.
And number of parameters is : $16 \times (5 \times 5 \times 3 + 1) = 1216$
(The 1 has been added for bias)

b)

After first layer's convolution the shape of our tensor is:

$$16 \times 128 \times 128$$

After first layer's pooling the shape of our tensor is:

$$16 \times 64 \times 64$$

After the second layer's convolution the shape of our tensor is:

$$16 \times 64 \times 64$$

After the second layer's pooling the shape of our tensor is:

$$16 \times 32 \times 32$$

After the third layer's convolution the shape of our tensor is:

$$16 \times 32 \times 32$$

After the third layer's pooling the shape of our tensor is:

$$16 \times 16 \times 16$$

Number of parameters equals:

$$16 \times (5 \times 5 \times 3 + 1) + 16 \times (5 \times 5 \times 16 + 1) + 16 \times (5 \times 5 \times 16 + 1) = 14048$$

c)

We will add a 1 layer fully connected network to the end of our previous network:

$$14048 + 10(16 \times 16 \times 16 + 1) = 55018$$

d)

using 5×5 convolution in three layers results in three rises in receptive field with size 2×2 around the first receptive field. Note that pooling layer does not change the receptive field.

first RF equals 1. So we have:

$$(1 + 2 \times 2 + 2 \times 2 + 2 \times 2) \times (1 + 2 \times 2 + 2 \times 2 + 2 \times 2) = 13 \times 13$$

So the receptive field after third layer is : 13×13

Question 3

part 1

a)

Encoder:
- Repeated application of two 3×3 Convolutions (unpadded)+ReLU
- 2×2 Max Pooling with stride 2

Decoder:
- 2×2 up-convolution
- Repeated application of two 3×3 Convolutions (unpadded)+ReLU
- Cropping: Loss of border pixel (see in/out size)
- 64 channel feature vector

Data Augmentation:
- Shift, Rotation, and Random Deformation

The U-Net architecture is a convolutional neural network (CNN) designed for semantic image segmentation, a task where the goal is to classify each pixel of an input image into different classes. The main idea behind U-Net is its unique architecture, which distinguishes it from other neural networks, particularly for image segmentation tasks.

Here are the key features that make U-Net distinctive(from the most important to the least):

1. Encoder-Decoder Structure: U-Net has a U-shaped architecture, consisting of an encoder and a decoder. The encoder captures context and extracts features from the input image through a series of convolutional and pooling layers. The decoder then upsamples the feature maps to generate a segmentation map.

2. Skip Connections: One of the critical innovations of U-Net is the use of skip connections. These connections directly link the corresponding layers between the encoder and decoder. This allows the network to retain high-resolution information from the encoder during the upsampling process in the decoder. Skip connections help in preserving spatial information and addressing the vanishing gradient problem during training.

3. Multi-Resolution Features: U-Net incorporates features at multiple resolutions in the decoder to combine both local and global contextual information. The skip connections facilitate the fusion of high-level and low-level features, enabling the network to make precise predictions while maintaining a global understanding of the input.

The U-Net architecture has proven to be effective for various medical image segmentation tasks, such as identifying organs or lesions in medical scans. Its ability to capture detailed spatial information and leverage skip connections for improved segmentation performance makes it a popular choice in the computer vision community, particularly for tasks involving pixel-level classification.

b)

Skip Connections: One of the critical innovations of U-Net is the use of skip connections. These connections directly link the corresponding layers between the encoder and decoder. This allows the network to retain high-resolution information from the encoder during the upsampling process in the decoder. Skip connections help in preserving spatial information and addressing the vanishing gradient problem during training.

c)

1. Preservation of Spatial Information: Medical images often contain fine structures and detailed patterns that are critical for accurate segmentation. Skip connections allow the model to retain high-resolution information from the encoder, preserving fine details that may be lost during the downsampling process. This is essential for maintaining spatial accuracy in medical image segmentation.

2. Handling Varied Anatomy and Pathology: Medical images can exhibit significant variations in anatomy and pathology across different patients and imaging modalities. Skip connections enable the model to capture both local and global contextual information, allowing it to adapt to diverse patterns and structures present in medical images.

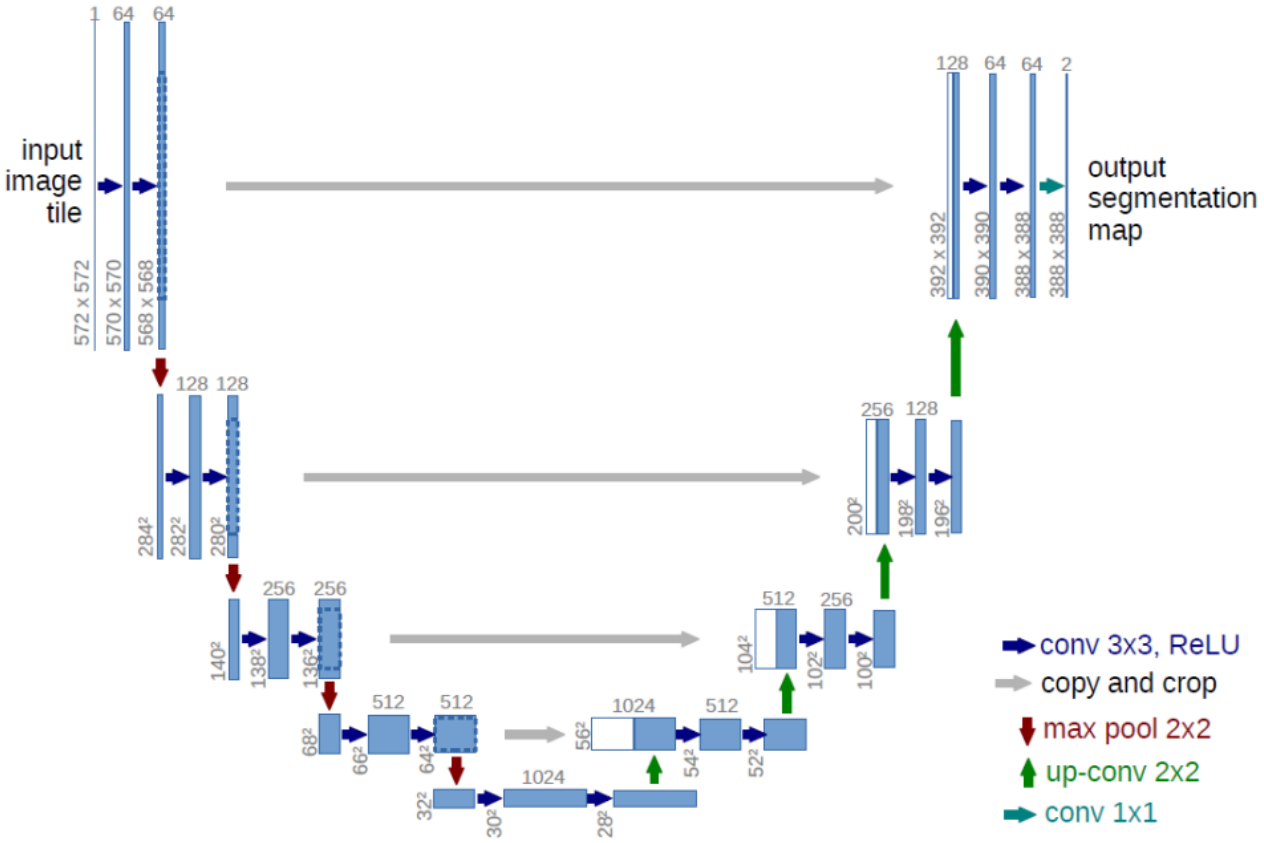
4. Enhancing Feature Fusion: Skip connections enable the fusion of features at multiple scales. The combination of both low-level and high-level features contributes to a more comprehensive understanding of the input image, facilitating accurate segmentation by capturing both global context and local details.

Overall, skip connections in U-Net and similar architectures contribute to the robustness and effectiveness of the model for medical image segmentation. The ability to preserve spatial information, adapt to diverse anatomies, and handle class imbalances makes skip connections a valuable component in neural networks designed for medical image analysis.

part 2

a)

With respect to the structure that exists in slides of our course and considering same convolutions:



$$256 \times 256 \rightarrow 128 \times 128 \rightarrow 64 \times 64 \rightarrow 32 \times 32 \rightarrow 16 \times 16$$

Note that number of channels will be found out by knowing number of filters.

b)

the input to the second layer is a 64-channel data. and it applies 128 filters to the input. So we have:

$$128 \times (64 \times 3 \times 3 + 1) = 73856$$

part 3

a)

In summary, while both DenseNet and ResNet aim to address challenges associated with training deep networks, DenseNet achieves this by densely connecting layers, promoting feature reuse and parameter efficiency, while ResNet uses residual connections to facilitate the training of very deep networks by addressing the vanishing gradient problem and promoting the learning of identity mappings.

Dense Connections in DenseNet:

Dense Connectivity: In DenseNet, each layer receives input not only from the previous layer but also from all preceding layers in the network. This creates densely connected blocks, where the output of each layer is concatenated with the inputs of all subsequent layers.

Feature Reuse: Dense connections facilitate better feature reuse, as each layer has access to the features learned by all its preceding layers. This can lead to more efficient parameter usage and improved gradient flow during training.

Parameter Efficiency: Dense connections reduce the number of parameters since each layer receives input from all preceding layers, allowing for more compact and expressive representations.

Residual Connections in ResNet:

Residual Blocks: In ResNet, each block of layers (typically containing two or three convolutional layers) has a "shortcut" or "skip connection" that bypasses one or more layers. The output of a layer is added to the input of one or more subsequent layers.

Identity Mapping: The idea is to learn the residual, the difference between the input and the output, making it easier for the network to learn the identity mapping. This helps with the training of very deep networks by mitigating the vanishing gradient problem.

b)

Short Paths for Gradient Flow:

In DenseNet, each layer receives input not only from the previous layer but also from all preceding layers in the block. This creates shorter paths for the gradient to flow during backpropagation.

Since each layer is connected to all preceding layers, the gradient has multiple routes to propagate through the network. This helps alleviate the vanishing gradient problem, as the gradients can take shorter paths and are less likely to diminish significantly.

Parameter Efficiency and Advantage in Computation:

Dense connections reduce the number of parameters in the network. Each layer receives input from all preceding layers, and the outputs of all these layers are concatenated.

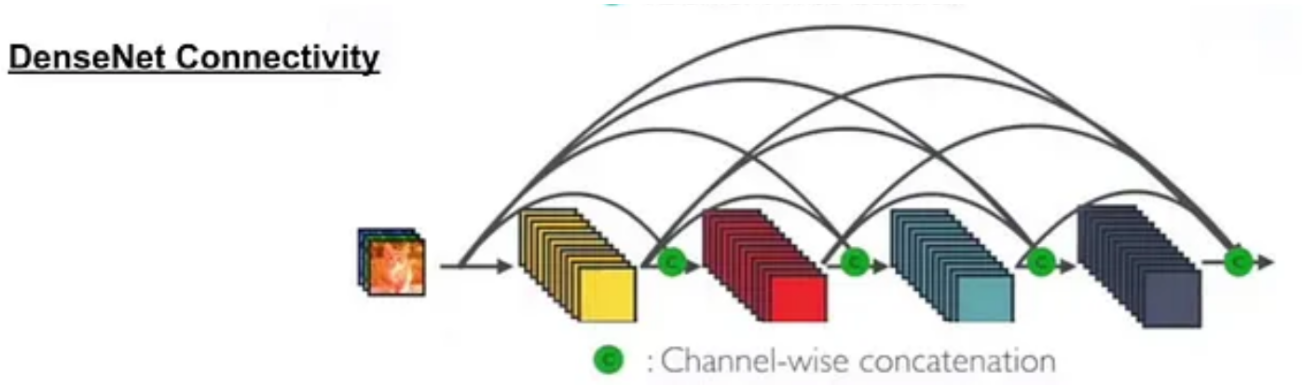
This parameter sharing allows for more efficient use of model parameters. Instead of learning redundant features independently in each layer, the network can focus on learning unique features at each layer while reusing features from earlier layers.

The dense connectivity in DenseNet contributes to better parameter efficiency, enabling the construction of deeper networks with fewer parameters compared to traditional architectures.

The dense connections facilitate parallelism in computation, as each layer's computation can be performed independently since it has access to the feature maps of all preceding layers.

part 4

a)



As you can see in the map above, the third layer gets input from first and second layer. So the number of input channels for third layer is:

$$64 + 128 = 192$$

b)

Inputs to second layer:

$$32 + 24 = 56$$

Inputs to third layer:

$$56 + 24 = 80$$

Output of third layer:

$$80 + 24 = 104$$