National University of Sciences and Technology

School of Electrical Engineering and Computer Science

Department of Computing

CS-405 Deep Learning

Fall 2023

# Lab 7

**Semantic Image Segmentation**

# Date: 27th Oct 2023

# Name: Alina Nasir

# CMS ID: 342350

# Instructor: Dr Daud Abdullah

**Question 1:**

**In the cell below, paths to the images and their corresponding masks are being set. Please analyze the mask of an image and check how does it store the information regarding classes (i.e. different parts) of a vehicle?**

**Answer:** The mask of an image stores information regarding the car in the form of tags like wheels, lights and windows. Each feature obtained from the car is assigned a tag and its co-ordinates are stored in a form of list corresponding to its tags.

**Question 2:**

**Question 2.1: Please observe the dataset class given in the cell below. When we want to retrieve items using an instance of this class, which function is generally called?**

**Answer:** The get\_item function is used when we want to retrieve items using an instance of this class. This function reads an image and its corresponding mask, applies any specified transformations, and returns the processed image and mask as a tuple when called with an index.

**Question 2.2: What does the data consist of that we obtain when we use an instance of this class for item retrieval?**

**Answer:** The data returned by the function consists of a tuple. The first element of the tuple contains the processed image data for the specific item at the given index. This image is typically in the form of a tensor and has undergone various transformations, including conversion to RGB color space, conversion to a PyTorch tensor, and normalization. The second element of the tuple contains the mask data associated with the image. In this code it is the grayscale mask corresponding to the image. It is also converted to a PyTorch tensor.

**Question 3:**

**In the cell below, from where did we pick up the values of mean and std of RGB values for input normalization?**

**Answer:** These values are chosen because they are roughly the mean and standard deviation of the ImageNet dataset, which is a large and diverse dataset commonly used for pre-training deep learning models. Normalizing new data with these values helps the model work well when fine-tuned on other tasks or when performing inference on new images.

**Question 4:**

**Which augmentations are we using in this case?**

**Answer:** In this code we are applying the following augmentations to the image.

1. Resize:

The images are resized to the target size of 256 x 256 pixels.

1. Horizontal Flip:

This applies random horizontal flip to augment the images in dataset by providing variations of the same image as if it were horizontally mirrored.

1. Vertical Flip:

This applies random vertical flip to augment the images in dataset by providing variations of the same image as if it were vertically mirrored.

1. Gaussian Noise:

This is added to introduce variability and robustness to the model during training.

**Question 5:**

**Please observe the SegNet model below and answer the following questions:**

**5.1: In SegNet we studied that during MaxPooling, the indices of the maxpooled values are stored during encoding stage and these values are then used in unpooling operations in the decoding stage. How is this being done in the code below? Which functions are being used in this process?**

**Answer:** The MaxPool2d function is used to perform the max pooling operation. During each encoding stage the max-pooling operation is performed the (return\_indices=True) argument allows the indices of the max-pooled values to be stored. These indices are needed for unpooling during the decoding stage.

In the decoding stages from stage 5 to 1, the MaxUnpool2d function is used to unpool the values. The MaxUnpool2d function takes the stored indices from the corresponding encoding stage and uses them to perform the unpooling operation. For example, in the Stage 5 decoding, the self.MaxDe(x, ind5, output\_size=size4) where ind5 contains the indices needed for unpooling, and size4 is the output size after the encoding stage. This process is repeated for each decoding stage.

**5.2: How many convolution blocks or stages are being used in the encoder and the decoder?**

**Answer:** The encoder is consisted of 5 stages. The stage 1 and 2 have 2 layers of Convolution, 2 layers of Batch Normalization and Max Pool respectively. The stage 3, 4 and 5 have 3 layers of Convolution, 3 layers of Batch Normalization and Max Pool respectively. Similarly, the decoder has also five stages where each stage corresponds to their respective counterparts in encoding.

**5.3: End the end of it all, what does the dim=1 in x = F.softmax(x, dim=1) signify?**

**Answer:** The dim=1 argument in F.softmax specifies the dimension along which the softmax operation is applied. In this code, it signifies that the softmax operation is applied along the second dimension of the tensor x. In the context of a segmentation task, where x is a tensor representing class scores for each pixel in an image, applying softmax with (dim=1) ensures that the softmax operation is computed separately for each pixel. It normalizes the scores for each class independently, and the output represents the probability distribution over the classes for each pixel.

**Question 6:**

**Please observe the UNet model below and answer the following questions:**

**6.1: What does the DoubleConv model do? It is built up using the Sequential method of building up models, what is this method?**

**Answer:** It is commonly used in various deep learning architectures for tasks such as image segmentation, classification, and object detection. This block consists of two consecutive convolutional layers, each followed by batch normalization and rectified linear unit (ReLU) activation. The first convolutional layer takes the input tensor with input channels and produces an output tensor with mid channels. It then applies batch normalization to the output of first layer which helps stabilize training by normalizing the inputs. A ReLU activation function is applied after batch normalization to introduce non-linearity. The second convolutional layer then takes the mid channels and produces the final output with output channels. It is also followed by batch normalization and ReLU activation.

**6.2: What operations does the Down module constitute? What is its basic purpose?**

**Answer:** The Down module constitutes operations for downscaling the spatial resolution of feature maps through max-pooling followed by a double convolution operation. Its basic purpose is to reduce the spatial dimensions of feature maps while increasing the number of channels to capture higher-level features. This is commonly used in U-Net-like architectures for image segmentation tasks.

**6.3: What is the purpose of the Up module? What operations does it consists of?**

**Answer:** The primary purpose of this module is to perform the up-sampling and combining of features in the decoder path of a U-Net architecture, facilitating the reconstruction of spatial details while preserving and combining important high-level features from the encoder path. It performs two operations. The first operation performs transposed convolution (also known as deconvolution or up-sampling) on the input feature map. It increases the spatial resolution by a factor of 2 and reduces the number of channels by factor of 2. It then applies double convolutional block defined in DoubleConv class.

**Question 7:**

**The segmentation\_models\_pytorch is a useful library for using pretrained models for the task of semantic segmentation. If we are using the Unet model, then in this case we can choose between different pre-trained models as the encoder for the UNet. In this cell below we have used "vgg16\_bn".Please visit the link given below and mention 3 other encoders listed in the repository.**

**Answer:** Three encoders mentioned are:

1. resnet18
2. densenet121
3. efficientnet-b0

**Question 8:**

**In the 2 cells given below, we are computing Pixel Accuracy as well as mean Intersection over Union evaluation metrics. What is the difference between the two? Which one is better and why?**

**Answer:** Pixel accuracy measures the proportion of correctly classified pixels in the segmentation. It provides a high-level view of the overall accuracy of the model whereas mIoU is a more detailed metric that assesses the quality of the segmentation by considering the overlap between predicted and ground truth regions for each class. Pixel accuracy is useful when you want to assess the overall performance of the model in terms of pixel-level classification accuracy whereas mIoU is useful when you want to evaluate the model's ability to correctly classify and segment objects in different classes, providing more insight into class-specific segmentation performance. Since we are performing specific segmentation of features of cars mIoU will be better in our case.

**Question 9:**

**If you had to use a pre-trained encoder other than the VGG16\_BN for the UNet model, which one would you prefer to use, and why? (Note: There is no single correct answer to this question. Follow your intuition and current knowledge base that we have developed in the classes.)**

**Answer:** I would have preferred ResNet18 encoder because ResNet-18 introduces the concept of residual connections, which allows for the training of very deep networks (hundreds of layers) without suffering from vanishing gradients. U-Net architectures benefit from deeper encoders because they can capture more complex and hierarchical features. ResNet-18 includes skip connections that can be leveraged in U-Net architectures for better information flow and feature fusion between encoding and decoding paths. ResNet-18 is computationally efficient compared to VGG16 with batch normalization, which has more layers and parameters.

**Question 10:**

**Why are we creating the CarsTestDataset class? Why not use the same dataset class that we had defined earlier? What is the difference between the two?**

**Answer:** The reason for creating the CarsTestDataset class is likely to handle the testing dataset separately from the training and validation datasets. While the CarDataset class is designed for the training and validation datasets, the CarsTestDataset class is tailored for the testing dataset. In the CarDataset we applied series of transformation and augmented the dataset horizontally, vertically and by Gaussian distribution.

**Comparison:**

The testing score achieved is 0.668 and the validation score is 0.7 which shows that this model is over-fitting the dataset.