# Very Deep Learners: Group02

## Assignment 3 - Submission:

**1.1 THEORY PROBLEMS:**

**1. How can a fully connected layer can be realized as a convolution layer ?**
A fully connected layer can only deal with fixed-size input space and “fully connects” the input and output”. A convolutional layer *slide* a filter across the input so it does not need size parameters.This fully connected layer works is that it looks at the output of the previous layer (which as we remember should represent the activation maps of high level features) and determines which features most correlate to a particular class. It can be realized as a convolution layer as As the filter is sliding, or **convolving**, around the input image, it is multiplying the values in the filter with the original pixel values of the image, and realizing the convolution layer. [4]

**2. What is the importance of skip connections in a CNN for image segmentation and object detection problems?**
Skip connections can skip multiple layers in the CNN and *skip* to connect to the next layer. For image segmentation and object detection, this allows us to **optimise**. Skip connections are said to “breaking the permutation symmetry of nodes, by reducing the possibility of node elimination and by making the nodes less linearly dependent”[1]. For image segmentation, there is potential for learning-slow down. Skipping connections moves us away from “ghosts”[1] and allows us to transfer to the next or next couple of layers without losing connection to the prior.

**3. Ground truth labels for image classification problem are class names which are converted to one hot vectors and cross-entropy loss is applied over CNN to train them in supervised manner. What are the ground truth labels in semantic segmentation problem, and by what loss function are they trained over a CNN?**
In semantic segmentation, ground truth labels are set by pixel. Each semantic pixel will have it’s own label, and the class is understood at the pixel level.
In our research, over a CNN, we saw cross-entropy loss function paired with a dice loss function. These two paired together seemed so show the best results. The Dice loss function is easily implemented and a nice addiction to the CNN for loss. [2] This can be extended to the Intersection over Union (IoU) metric.

**4. How transposed convolution helps in upsampling an image, in case of semantic segmentation ?**

Upsampling is used to generate low resolution images to high resolution. The transposed convolution is which is the process of going in the opposite direction of a normal convolution. This is done by maintaining the connectivity pattern and moving through the image. As adjusting pixels, we can effectively *upsample,* or move from smaller input to larger input automatically. The transposed convolution helps upsample **optimally** for semantic segmentation as it does not use predefined interpolation method, but rather learnable parameters.[4]

**5. Why accuracy is not a good measure in case of semantic segmentation ? What measure is used to evaluate results in semantic segmentation ?**

Accuracy may not be a good measure as it does not tell us the pixel-by-pixel accuracy of true-positive, vs. false-positive etc.  Instead, measures which focus on the pixel-accuracy or ratio of the **detection area compared to the ground truth area** are better suited.

1. **Pixel accuracy:** binary mask -- a true positive represents a pixel that is correctly predicted to belong to the given class (according to the target mask) [3]
2. **Intersection over Union** (number of pixels common between the target and prediction masks divided by the total number of pixels present across both masks) [3]
3. **Average Precision:** compare predicted masks with each of the available target masks for a given input [3]

Sources:

1[https://arxiv.org/abs/1701.09175]
2[https://arxiv.org/pdf/1511.00561.pdf]
3[https://www.jeremyjordan.me/evaluating-image-segmentation-models/]
4 [https://towardsdatascience.com/up-sampling-with-transposed-convolution-9ae4f2df52d0]

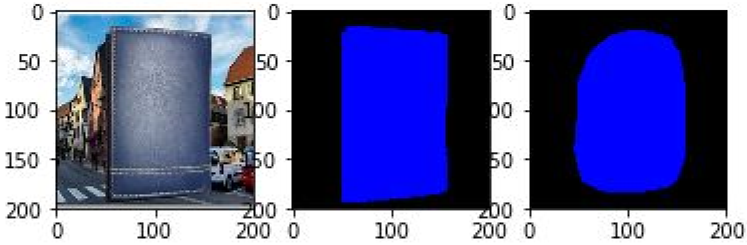
---

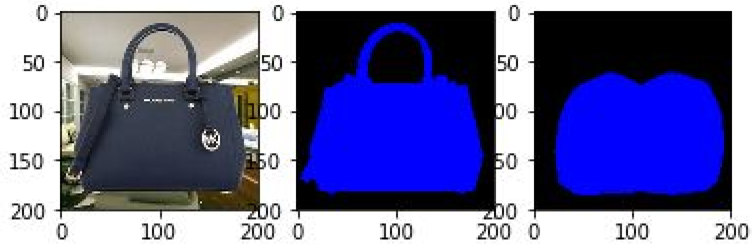
**1.2 PRACTICAL PROBLEMS:**

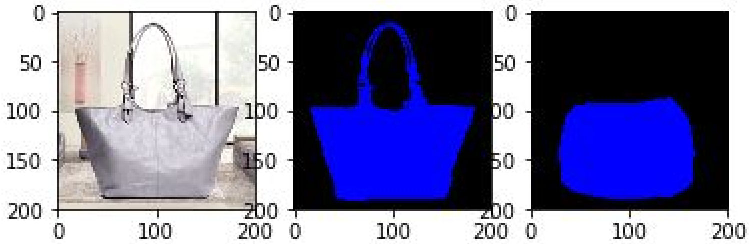
1. After completing the code, the FCNs32 model gave the results at 8000 iterations (18 epochs).

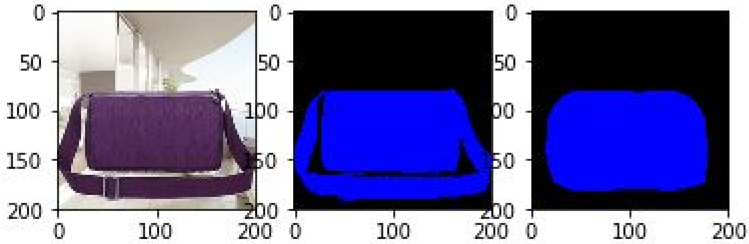
| | |
|---|---|
| Epoch 1 - 18 | **mean iou score/val**<br><br>0.316749356<br>0.661428185<br>0.670798589<br>0.658773262<br>0.635150582<br>0.675304538<br>0.65913212<br>0.660272587<br>0.673351826<br>0.662667339<br>0.673732397<br>0.670394612<br>0.676688951<br>0.677283299<br>0.674414111<br>0.665548775<br>0.673324803<br>0.673934933<br>0.662125066<br>0.316749356<br>0.661428185<br>0.670798589<br>0.658773262<br>0.635150582<br>0.675304538<br>0.65913212<br>0.660272587<br>0.673351826<br>0.662667339<br>0.673732397<br>0.670394612<br>0.676688951<br>0.677283299<br>0.674414111<br>0.665548775<br>0.673324803<br>0.673934933<br>0.66212506 |

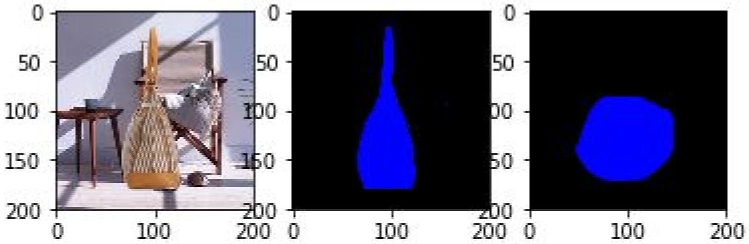
## 2. 5 Predictions:











2. After completing the code, the FCNs8 model gave the results at 8000 iterations (18 epochs). The results at the final iteration were better, reaching **0.7959**.

| | |
|---|---|
| Epoch 1 - 18 | **mean iou score/val**<br>0.316749356<br>0.662014521<br>0.658925963<br>0.673922757<br>0.68321448<br>0.682177557<br>0.700641713<br>0.700251964<br>0.720603671<br>0.727878855<br>0.735223603<br>0.723733819<br>0.751845672<br>0.747673587 |

| | |
|---|---|
| | 0.769901563<br>0.763286321<br>0.77206065<br>0.770117971<br>0.79583449 |

## 2.1 THEORY PROBLEMS:

1) **What is a receptive field of a convolution filter?**
The receptive field of a neuron is its covered region in the image plane. The receptive field in CNN is a hyper parameter that defines the area of the input space that affects a particular unit of the network. Here the input space might be the output of some layer and an input to the next layer. Since it is impractical to connect a unit to all previous units, each unit in a hidden layer is only connected to a number of units in the previous layer. This small patch of region is the receptive field.

2) **How can a receptive field of a filter of a particular layer be obtained in theory ?**
To obtain the receptive fields in theory, we should determine segmentation of edges and regions. Then, remove those segments from image iteratively. At each iteration, we should remove the segment of the image that results in the smallest decrease of the classification score. In each iteration, we slowly remove parts of image until the image is incorrectly classified. At the end, the image we have contains the most important pixels of the original image. In any layer, the original image is the output of previous layer, so that we can find the receptive field of different filters in each layer.

3) **How can a receptive field of a filter can be obtained in practice ?**
To obtain actual or empirical receptive fields, the data driven approach is used. We should select top K images with the highest activations for a given unit. We need to identify exactly which regions of the image lead to high unit activations. To do this, we should use replicated images with small random occluders in different locations. This results in a number of occluded images per original image. After having all different occluded images, we feed all occluded images to the network and look at the change in the activations. If there is a big inconsistency(discrepancy) in change of activation compared to using the original image, it means the given patch is important. Because, the occluders we used highly impacted the activation which means that region is important for that feature. As a last step, to consolidate the information from K images, we choose the maximumly activated unit and center our final receptive field around spatial location of that unit. We take an average of the inconsistency(discrepancy) map around the center of final receptive field and we have our final receptive field.

**2.2**

1. **Send the visualization of 10th unit(filter) of conv layer 4 of Resnet 18**

The 10th unit is visualized below after running the visualization code.