

CSCE 689 - Deep Learning for Computer Graphics

Programming Assignment 2

Deadline: Oct. 25th

In this assignment, we use a convolutional neural network (CNN) to increase the resolution of images by a factor of 2. We perform this process in two different ways. First, we directly perform the super-resolution on the images in the RGB color space. In the second part, we use the Lab color space to increase the resolution of the input images. In both parts, we use the network, shown in Fig. 1, to handle the application.

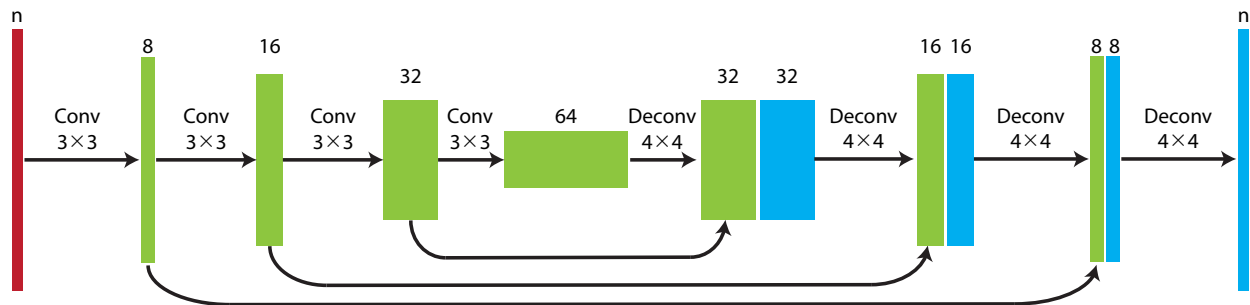


Figure 1: The network contains four convolutional layers in the encoder, while it has four deconvolutional layers in the decoder. The convolutional and deconvolutional layers have a kernel size of 3×3 and 4×4 , respectively. Both convolutional and deconvolutional layers have a stride of 2 and all are followed by rectified linear unit (ReLU) except for the last one, which is sigmoid. The number of input and output channels are denoted by n .

1 Super-Resolution in the RGB Space (9 points)

Follow the below steps to set up the model and train the network.

- We use the STL10 dataset, containing images of size 96×96 , for training and testing the network. You can load this dataset using a similar function to the one used to load the CIFAR 10 dataset in the previous assignment. Load the “Unlabeled” dataset (100,000 images) as the training data and “Test” dataset (8,000 images) as the testing data.
- Create the network in Fig. 1. In this part of the homework the number of input and output channels are three.
- At this stage, you need to setup the input and the ground truth images. The ground truth image is the one you fetch from the dataset. To create the input image, use the `nn.functional.interpolate` to downsample the image by a factor of 2 using the “bilinear” method. This process gives us an image of size 48×48 . The final input image is obtained by upsampling this image by a factor of 2. Note that, the downsampling and upsampling steps basically remove the spatial details in the image. Now, we would like to recover the missing details by the network.
- Use MSE loss with Adam optimization method to train the network. Set the mini-batch size equal to 5 and perform the training for 5 epochs.
- Evaluate the error on the first 100 images of the training set at every 500 iterations of training.

Report the followings:

1. A plot showing the test error as a function of the training iterations.
2. Show the output of the network for the first five images in the test set. Also show the corresponding ground truth as well as the input images.

2 Super-Resolution in the Lab Space (6 points)

Here, we perform the process in the Lab color space and only on the L channel. The network in this case, takes the L channel of the input image and output the super-resolved L channel. This is compared to the ground truth L channel during training. So instead of super-resolving the image in the RGB space, we only use the network to super-resolve the L channel. The final image is then reconstructed by combining the L channel from the network with the ab channels from the input image. Note that, the number of input and output channels of the network in this case are equal to 1, i.e. $n = 1$ in Fig. 1.

In order to convert the images from RGB to Lab color space, use the `color.rgb2lab` function from `skimage` toolbox. You have to first transfer the tensor to numpy array, do the conversion, and transform it back to tensor. Note that the range of values in the L channel are from 0 to 100. So before passing the L channel as the input you need to divide it by 100. You have to do the same to the L channel of the ground truth image when you are computing the MSE loss. This ensures that the output of the network is in the range 0 to 1.

Report the followings:

1. A plot showing the test error as a function of the training iterations. Note that, you have to compute the test error on the RGB images. So just combine the output of the network (the L channel) with the ab channels from the input and convert it from Lab to RGB. Note that you need to multiply the output of the network by 100 before combining it with the ab channels.
2. Show the output of the network for the first five images in the test set. Also show the corresponding ground truth as well as the input images.

3 Deliverables

- Please submit the codes for the two parts of the assignment separately. To make things easier, please use the following filenames for each part: `RGB.py` for the first part and `Lab.py` for the second part.
- Please submit a report along with the codes containing the requested items in the boxes in pdf format.
- Please put all the codes and the report in a folder and zip them up. The zip file should be named “`firstname.lastname.zip`”. Submit the zip file through e-campus.