

Assignment Three

Instructions:

Programming Part:

- Submit your code along with the results obtained from running it.
- Two .ipynb files, colour_regression.ipynb and colourization.ipynb, have been provided to assist with the implementation.
- However, it is highly recommended to attempt the implementation independently before referring to the provided code.

Theory Part:

- Submit your answers, observations, and findings in PDF format.

Convolutional Neural Network for Image Colorization

In this assignment, we will train a convolutional neural network to address the challenge of image colorization. The objective is to predict the color of each pixel in a given grayscale image. Image colorization is inherently complex, primarily because it is an ill-posed problem—a single grayscale image can correspond to multiple equally valid color representations.

Dataset: For this task, we will use the CIFAR-10 dataset, which consists of 32×32 pixel images. To simplify the problem, we will focus only on the "Horse" category from the dataset.
Now, let's begin colorizing some horses!

Colourization as Regression:

There are multiple ways to approach image colorization as a machine learning task. One straightforward method is to treat it as a regression problem, where the objective is to predict RGB intensities for each pixel based on the grayscale input. Since the outputs are continuous, the model can be trained using squared error loss.

Questions:

Convolution Layers:

- How many convolution layers are present in the regression-based CNN model?
- What are the filter sizes and the number of filters in each layer?
- Represent this information using a table or diagram for clarity.

Training Epochs:

- How many epochs was the CNN model trained for?

Training Variations:

- Train multiple models using different numbers of epochs.
- Analyze how output images and training loss change when the number of epochs is increased or decreased.
- Discuss the impact of training duration on the model's performance.

Colourization as Classification:

In this approach, we treat image colorization as a pixel-wise classification task by selecting a subset of 24 colors and assigning each pixel to one of these predefined colors. These 24 colors were obtained using k-means clustering, and the cluster centers are stored in `colour_kmeans24_cat7.npy`.

Building a Classification-Based CNN for Colorization:

- Design a CNN model for colorization, following the same layer structure and convolutional filters as the classification-based CNN, except for the output layer.
- Utilize PyTorch layers such as `nn.ReLU`, `nn.BatchNorm2d`, and `nn.MaxPool2d`.
- Instead of using `nn.Conv2d`, implement the custom convolution layer `MyConv2d`, provided in the code, to gain a deeper understanding of convolutional operations.

Generating and Comparing Images:

- Generate sample images using the trained classification-based model and compare them with those produced by the regression-based model.
- Analyze the differences in the outputs.
- Did the classification-based model result in better validation loss and accuracy?
- Discuss the qualitative differences in the generated images and provide an in-depth analysis of the results.

Skip Connections:

A skip connection in a neural network allows information to bypass one or more layers, directly connecting to a later layer. In this assignment, we will incorporate skip connections into our model.

Adding Skip Connections:

- Implement skip connections such that the first layer connects to the last, the second layer connects to the second last, and so on.
- As a result, the final convolution layer will receive both the output from the previous layer and the original grayscale input.
- This type of skip connection, commonly known as a "UNet", was introduced by [1].

Training the "UNet" Model:

- Train the UNet model for the same number of epochs as the previous CNN, using a batch size of 100.
- Plot the training curve and compare the results with the previous model.

Address the following:

- Did the skip connections improve the validation loss and accuracy?
- Did the skip connections lead to qualitatively better outputs? If so, how?
- Provide at least two reasons why skip connections may enhance the performance of CNN models.

Visualizing Intermediate Activations:

We will analyze the intermediate activations for various inputs by executing the visualization block in the colourization.ipynb notebook, which has been pre-written for your convenience.

Visualizing CNN Activations for Test Examples:

- Describe how the activations in the early layers differ from those in the deeper layers of the CNN. You do not need to include the output images in your write-up—only a detailed description of your observations.

Visualizing Colorization UNet Activations for Test Examples:

- Discuss the differences between the activations in the UNet and those observed in the CNN. Provide a detailed analysis based on your observations, without including the output images.

Some More Theoretical Questions:

1. The loss functions and evaluation metrics provided in the supplementary code are defined at the pixel level. However, pixel-level measures often do not align well with human perception of visual quality. How can the evaluation process be improved to better reflect human judgment? Reference [2] may provide insights into addressing this issue.
2. Our convolutional neural networks were trained on 32×32 input and output images. However, during testing, the desired output size is often different. Explain how the trained models can be adapted to colorize test images larger than 32×32.

References:

- [1]. Ronneberger, O., Fischer, P., and Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 234-241). Springer, Cham.

[2]. Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." In European conference on computer vision, pp. 694-711. Springer, Cham, 2016.