|  |  |
| --- | --- |
| **Ex No: 6.4**  **Date: 11/09/2024** | **Lab Record: Denoising with a CNN Autoencoder** |

**Objective:**

The objective of this lab is to train a convolutional neural network (CNN) autoencoder to denoise images from the Fashion MNIST dataset. By adding random noise to the dataset images and training the autoencoder, the model learns to reconstruct clean images from noisy inputs.

**Description:**

Autoencoders are neural networks designed to learn efficient representations of input data. In this lab, a CNN-based autoencoder is used to remove noise from images. The dataset used is Fashion MNIST, which consists of 28x28 grayscale images of clothing items. The autoencoder model has an encoder, a bottleneck, and a decoder:

* **Encoder**: Compresses the input images into a lower-dimensional feature space using convolutional layers.
* **Bottleneck**: The intermediate representation that holds the most relevant features of the input image.
* **Decoder**: Reconstructs the image from the compressed representation back to its original size using upsampling layers.

The noisy images are generated by introducing random Gaussian noise to the original images. The model is trained to map noisy images to clean ones.

**Explanation of Code:**

1. **Imports:** TensorFlow and TensorFlow Datasets are imported for model building and data handling. The matplotlib library is used for visualizing the model's outputs.
2. **Dataset Preparation:** The Fashion MNIST dataset is loaded and split into training and testing sets. Random noise is added to the images, and both noisy and clean images are normalized to values between 0 and 1. The function map\_image\_with\_noise generates noisy images that will serve as inputs to the model. The dataset is then shuffled, batched, and prepared for training.
3. **Model Architecture:** The autoencoder consists of three parts: the encoder, bottleneck, and decoder.
   * **Encoder:** Two convolutional layers are used with 64 and 128 filters, followed by max pooling layers to downsample the image.
   * **Bottleneck:** A convolutional layer with 256 filters is used to create a compressed representation of the input. A separate convolutional layer with a sigmoid activation is used to visualize the encoded output.
   * **Decoder:** The decoder uses convolutional layers and upsampling layers to reconstruct the image back to its original size.
4. **Training:** The model is compiled with the Adam optimizer and binary cross-entropy loss, and trained for 40 epochs using the noisy input images and clean target images.
5. **Results Visualization:** After training, sample noisy input images are passed through the autoencoder to generate the denoised output. The results are displayed, showing the noisy input, encoded features, and reconstructed images.

This lab demonstrates how a CNN autoencoder can be used for denoising tasks by learning to reconstruct clean images from noisy inputs.

**GitHub Link:** https://github.com/dyuthiramesh/Deep\_Learning\_Elective/blob/main/Sem5/Lab5/cnn\_transfer\_learning\_DISTRI.ipynb