|  |  |
| --- | --- |
| **Ex No: 7**  **Date: 11-09-2024** | **Denoising with a CNN Autoencoder** |

**Objective:**

The objective of this lab is to implement a convolutional neural network (CNN) autoencoder for denoising images from the Fashion MNIST dataset. By adding random noise to the images, the model will be trained to reconstruct the original, clean images, learning to remove the noise effectively. The focus is on building an autoencoder with an encoder, bottleneck, and decoder, which encodes the input images into a latent representation and then decodes them back to the original image size while minimizing the noise.

**Description:**

We utilize the Fashion MNIST dataset to demonstrate the capabilities of an autoencoder in image denoising. The dataset is prepared by normalizing the pixel values and introducing random Gaussian noise to each image. The autoencoder consists of three main components:

1. **Encoder**: This part of the model compresses the input images through a series of convolutional layers and max pooling operations. The encoder learns to capture important features of the images in a lower-dimensional latent space.
2. **Bottleneck**: The bottleneck layer represents the compressed latent space where the most critical features of the images are stored. This layer plays a key role in minimizing noise.
3. **Decoder**: The decoder upsamples the latent space back to the original image size, reconstructing the input. It uses upsampling layers to reverse the compression process while removing noise.

**Model Architecture:**

1. **Input Layer**:

* Input shape: (28, 28, 1) – Grayscale images from the Fashion MNIST dataset.

2. **Encoder**:

* **Conv2D**: 64 filters, kernel size (3x3), ReLU activation, padding='same'
* **MaxPooling2D**: Pool size (2x2)
* **Conv2D**: 128 filters, kernel size (3x3), ReLU activation, padding='same'
* **MaxPooling2D**: Pool size (2x2)

3. **Bottleneck**:

* **Conv2D**: 256 filters, kernel size (3x3), ReLU activation, padding='same'
* **Conv2D (Visualization)**: 1 filter, kernel size (3x3), Sigmoid activation, padding='same'

4. **Decoder**:

* **Conv2D**: 128 filters, kernel size (3x3), ReLU activation, padding='same'
* **UpSampling2D**: Size (2x2)
* **Conv2D**: 64 filters, kernel size (3x3), ReLU activation, padding='same'
* **UpSampling2D**: Size (2x2)
* **Conv2D (Output)**: 1 filter, kernel size (3x3), Sigmoid activation, padding='same'

5. **Output Layer**:

* Reconstructed image of shape (28, 28, 1) (same as the input).

**Loss Function and Optimizer**

* **Loss Function**: The Mean Squared Error (MSE) loss is typically used for autoencoders because the goal is to minimize the difference between the reconstructed (denoised) image and the original image.
* **Optimizer**: The Adam optimizer is commonly chosen for this task due to its efficiency in handling large datasets and its adaptive learning rate.

**Github Link:**

**https://github.com/Bhargava-Srinivasan-26/Deep\_learning\_elective/tree/main/Unit%202/Lab%206.4**