| Experiment No. 5 |
| --- |
| Design the architecture and implement the autoencoder model for Image denoising. |
| Date of Performance: |
| Date of Submission: |

**Aim:** Design the architecture and implement the Autoencoder model for Image Denoising.

**Objective:** Ability to design an Autoencoder model capable of removing noise from input images and reconstructing the original clean images effectively.

**Theory:**

Autoencoders are a class of **self-supervised neural networks** used to learn efficient representations of data, typically for the purpose of **dimensionality reduction**, **feature learning**, or **data reconstruction**.
They consist of two main parts:

- **Encoder:** Compresses the input image into a lower-dimensional latent representation. It captures the most important features while discarding noise and redundancies.

- **Decoder:** Reconstructs the image back from the latent representation, ideally reproducing the clean version of the original image.

In **Image Denoising**, the Autoencoder learns to map **noisy images → clean images**. During training, the model is provided with noisy inputs but clean targets. The model then learns to ignore irrelevant noisy information and retain the underlying image structure.

A **Convolutional Autoencoder (CAE)** is particularly effective for image denoising because convolutional layers can capture spatial hierarchies and local features of the image.

## Working Principle:

1. Add random noise (Gaussian or salt-and-pepper) to clean images.
2. Feed the noisy images to the Autoencoder.
3. The Encoder compresses and removes noise features.
4. The Decoder reconstructs clean images from the compressed latent representation.

5. The model is trained using a reconstruction loss function such as **Mean Squared Error (MSELoss)** to minimize the difference between clean and reconstructed images.

**Applications of Autoencoders in Image Denoising:**

1. **Noise Reduction:** Removes unwanted noise from scanned documents, photographs, and medical images.
2. **Preprocessing Step:** Used in computer vision pipelines to improve input quality before further analysis.
3. **Restoration:** Helps restore old or degraded images by reducing artifacts and distortions.

**Conclusion:**

The designed Convolutional Autoencoder effectively reduced noise from images, showcasing the model's ability to learn essential image features while filtering out irrelevant noisy data. This experiment demonstrates that Autoencoders can serve as powerful unsupervised feature extractors and are valuable in applications such as image restoration, noise reduction, and feature compression.

**1. Summary of the Experiment**

The experiment involves a **Convolutional Autoencoder** trained on noisy images. The model's objective was to **reconstruct clean images** from noisy inputs by learning essential features. Key observations:

- The model successfully **reduced noise**, demonstrating effective denoising capability.
- It captured **core features** of the images, while discarding irrelevant noise.
- Training was unsupervised, relying only on input images and reconstruction targets.

---

**2. Critical Analysis**

**Merits:**

1. **Effective Noise Reduction:**
    - The CAE learned to suppress irrelevant pixel-level noise while retaining important structures.

   - Particularly useful for preprocessing images in downstream tasks (classification, segmentation).
2. **Unsupervised Feature Learning:**
   - Does not require labeled data.
   - Extracted compressed representations (latent space) that can serve as input features for other models.
3. **Applications:**
   - **Image Restoration:** Reconstructing corrupted or low-quality images.
   - **Noise Reduction:** Cleaning medical images, satellite images, or surveillance footage.
   - **Dimensionality Reduction / Feature Compression:** Useful for storage and faster computation.

**Limitations / Challenges:**

1. **Over-smoothing:**
   - Sometimes CAEs can over-smooth images, losing fine details in the process.
2. **Dependence on Dataset:**
   - Performance heavily depends on the diversity and quality of the training images.
   - If the noise type changes (e.g., Gaussian vs. Salt-and-Pepper), retraining may be necessary.
3. **Computational Cost:**
   - Training deep CAEs can be resource-intensive, especially for large images or high-resolution datasets.
4. **Generalization:**
   - The model may perform well on seen noise patterns but may struggle with unseen or extreme noise types.