Programming task

Instructions:

1. **Dataset Preparation:**
    - Select an image dataset suitable for the task (e.g., MNIST, CIFAR-10).
    - Preprocess the images (e.g., normalization) and introduce random noise to create noisy versions of the images.
2. **Model Architecture:**
    - Design a basic autoencoder architecture with an encoder and decoder component.
    - Implement denoising by adding noise to input images and reconstructing clean versions as output.
3. **Model Training:**
    - Compile the autoencoder model with an appropriate loss function (e.g., mean squared error) and optimizer (e.g., Adam).
    - Train the model on a set of noisy images, aiming to reconstruct clean versions of the input images.
    - Monitor training/validation loss to ensure proper convergence.
4. **Model Evaluation:**
    - Evaluate the trained model on a separate set of noisy images using relevant evaluation metrics such as mean squared error or structural similarity index (SSIM).
    - Visualize the reconstructed images and compare them with the ground truth to assess the model's performance.
5. **Hyperparameter Tuning:**
    - Experiment with different hyperparameters such as learning rate, batch size, and sparsity regularization parameters.
    - Use techniques like grid search or random search to find the optimal set of hyperparameters.
6. **Feature Engineering (Optional):**
    - Explore additional architectural variations or regularization techniques to improve the autoencoder's performance.
    - Consider incorporating convolutional layers for handling spatial information in the image data.
7. **Discussion and Analysis:**
    - Discuss the challenges encountered during model training and optimization, especially regarding sparsity constraints and denoising.
    - Analyze the quality of reconstructed images and the effectiveness of the autoencoder in removing noise and preserving image features.

- Reflect on potential improvements or alternative approaches for enhancing reconstruction performance.