Image Denoising Project Report

**Introduction**

Image denoising is a critical task in image processing and computer vision, aimed at improving the quality of images by removing noise. Noise can be introduced during image acquisition due to various factors like poor lighting conditions, sensor inaccuracies, or high ISO settings. In this project, we developed a model that takes noisy images and converts them into denoised images using a simple autoencoder architecture. This autoencoder encodes the noisy input images into a latent space and then decodes them back into denoised images. We trained our model using a specific training dataset and evaluated its performance on a test dataset.

**Dataset**

Our dataset is divided into two parts: training and testing.

- **Training Dataset :**This consists of pairs of noisy (low-quality) images and their corresponding denoised (high-quality) images.

- **Testing Dataset:**This contains noisy images and their corresponding ground truth denoised images.

The noisy images in both datasets are referred to as "low" images, while the clean, denoised images are referred to as "high" or "ground" images in the training and testing datasets, respectively.

**Preprocessing**

Before feeding the images into the model, we applied several preprocessing steps:

1. **Resizing:** We resized all images to a consistent size of 600x400 pixels.

2. **Grayscale Conversion:** Each image was converted to grayscale mode ('L').

3. **Rescaling:** The pixel values of the images were rescaled to the range of 0 to 255.

4. **Reshaping**: Finally, the images were reshaped to the dimensions (400, 600, 1), suitable for input into the model.

**Model Architecture**

The core of our image denoising system is an autoencoder composed of two main parts: the encoder and the decoder.

**Encoder**

The encoder compresses the input image into a lower-dimensional latent space. It consists of:

1. **First Convolutional Layer:**

- 64 filters with a kernel size of (3,3).

- Followed by a max-pooling operation with a pool size of (2,2).

2. **Second Convolutional Layer:**

- 32 filters with a kernel size of (3,3).

- Followed by another max-pooling operation with a pool size of (2,2).

**Decoder**

The decoder reconstructs the denoised image from the encoded latent space. It includes:

1. **First Upsampling Layer:**

- An upsampling operation to increase the dimensions of the encoded representation.

2. **Second Upsampling Layer:**

- Another upsampling operation to further increase the dimensions.

3. **Convolutional Layer:**

- A convolutional layer with 32 filters and a kernel size of (3,3).

4. **Output Convolutional Layer**:

- The final convolutional layer with 1 filter and a kernel size of (3,3), producing the denoised image.

**Compilation and Training**

The model was compiled using the Adadelta optimization algorithm and binary cross-entropy loss function. We split the training data into training and validation sets and employed early stopping to prevent overfitting.

**Evaluation**

To evaluate the performance of our model, we used the trained model to predict the denoised images from the noisy images in the test dataset. The performance was measured using the Peak Signal-to-Noise Ratio (PSNR), a widely used metric in image processing to assess the quality of reconstructed images.

The mean PSNR value obtained for the test dataset was approximately 17.

**Model Performance**

The autoencoder model was able to learn and identify patterns to some extent, as indicated by the PSNR value. However, a PSNR of 17 suggests that there is significant room for improvement. Typically, higher PSNR values (30 and above) are desirable for high-quality denoising. The moderate PSNR achieved can be attributed to several factors, including the complexity of the noise, the simplicity of the autoencoder architecture, and the optimization parameters used.

**Challenges and Limitations**

1. **Noise Complexity**: The noise in the images might have complex patterns that a simple autoencoder struggles to remove effectively.

2. **Model Simplicity:** The autoencoder architecture used in this project is relatively simple. More complex architectures, such as deeper convolutional neural networks or the use of advanced techniques like residual connections, could potentially yield better results.

3. **Optimization Parameters**: The choice of the Adadelta optimizer and binary cross-entropy loss function might not be the most optimal for this specific task. Experimenting with different optimizers (e.g., Adam) and loss functions (e.g., mean squared error) could improve performance.

**Future Work**

To enhance the model's performance, several improvements and extensions can be considered:

1. **Advanced Architectures**: Implementing more sophisticated models like U-Net, residual networks, or GANs (Generative Adversarial Networks) specifically designed for image denoising.

2. **Data Augmentation**: Applying various data augmentation techniques to increase the diversity and size of the training dataset, which could help the model generalize better.

3. **Hyperparameter Tuning**: Conducting extensive hyperparameter tuning, including experimenting with different optimizers, learning rates, batch sizes, and loss functions.

4. **Multi-Channel Images**: Extending the model to handle multi-channel (RGB) images instead of grayscale, which could provide more information for the denoising process.

**Conclusion**

In this project, we developed a simple autoencoder-based model for image denoising. The model was able to reduce noise to some extent, achieving a mean PSNR value of 17 on the test dataset. While this performance is moderate, it highlights the potential of autoencoder-based approaches for image denoising. Future work will focus on improving the model's architecture, optimizing the training process, and exploring advanced techniques to achieve higher quality denoised images.