Report on Image Denoising using U-Net Architecture

Written by: Kaustubh Dwivedi
22117066
Kaustubh_d@me.iitr.ac.in
Mechanical engineering IIT roorkee

Introduction

Image denoising is a crucial pre-processing step in many computer vision applications, aimed at removing noise from images while preserving important details. In this project, we implemented and trained a U-Net architecture to perform image denoising on a dataset of images.

Dataset Preparation

Data Extraction

The images were provided in a compressed file, `train.zip`, which was extracted to obtain the image dataset.

Dataset Splitting

The dataset was split into three parts:

- **Training Set**: 50% of the total images
- **Validation Set**: 20% of the total images
- **Test Set**: 30% of the total images

Data Augmentation

To train the model more effectively, each image was augmented by adding Poisson noise to create a noisy version of the image. This noisy image served as the input to the model, while the original clean image served as the target output.

Model Architecture

U-Net Architecture

The U-Net model was chosen for this task due to its effectiveness in image-to-image translation tasks. The architecture comprises an encoder and a decoder with skip connections that help in retaining high-resolution features throughout the network.

- **Encoder**: The encoder consists of convolutional blocks that reduce the spatial dimensions of the input image while increasing the depth (number of feature maps).
- **Decoder**: The decoder consists of transposed convolutional blocks that increase the spatial dimensions back to the original image size.
- **Skip Connections**: These connections pass feature maps from the encoder to the decoder, which helps in better reconstruction of the image.

Training Procedure

Hyperparameters

- **Learning Rate**: 0.001
- **Optimizer**: Adam optimizer
- **Loss Function**: Mean Squared Error (MSE) loss
- **Epochs**: 3000

Training Loop

The training loop involved the following steps:

1. **Forward Pass**: The noisy image was passed through the U-Net model to obtain the denoised output.
2. **Loss Calculation**: The loss between the output and the true image was calculated using the MSE loss function.
3. **Backward Pass**: The gradients were computed via backpropagation.
4. **Weights Update**: The model weights were updated using the Adam optimizer.

Validation

Every 100 epochs, the model was evaluated on the validation set to monitor its performance. The validation loss was computed and intermediate results were displayed using OpenCV.

Results

Training and Validation Loss

- The training and validation loss values were tracked and logged throughout the training process.

PSNR Calculation

- The Peak Signal-to-Noise Ratio (PSNR) was calculated on the test set to evaluate the performance of the denoising model.

Intermediate Outputs

- Intermediate outputs were displayed to visually assess the performance of the model during training.

Final Results

- **Training Loss**: *<0.0266 >*
- **Validation Loss**: *<0.021 >*
- **PSNR on Test Set**: *<27.68 >*

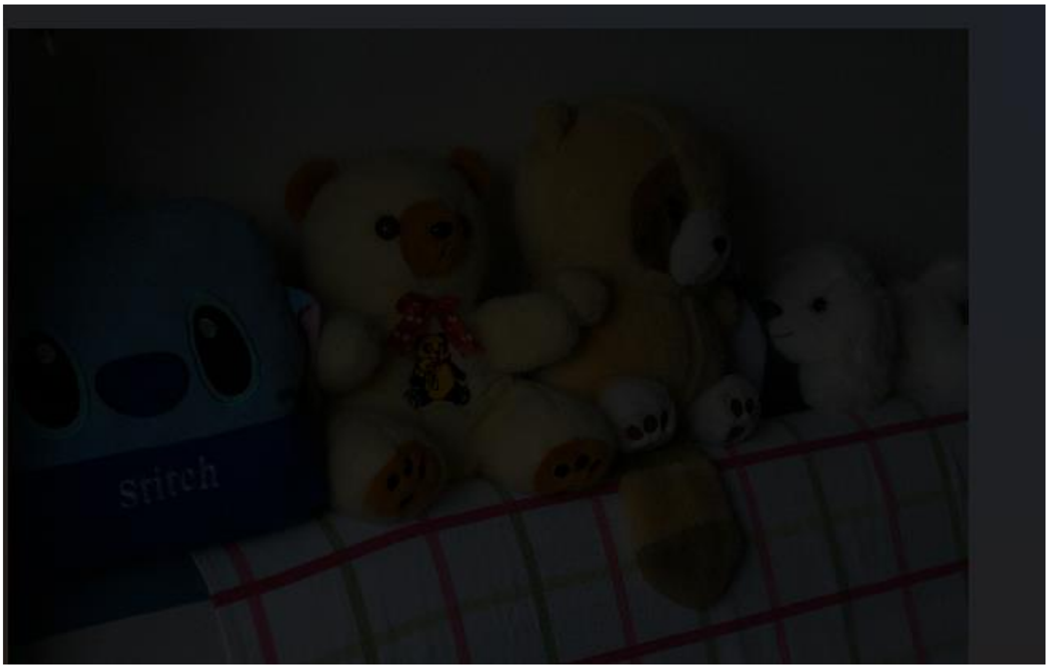
Discussion

Performance Analysis

- The model demonstrated a steady decrease in training and validation loss, indicating that it successfully learned to denoise the images.
- The final PSNR value on the test set indicates the quality of the denoised images. A higher PSNR value typically corresponds to better image quality.

Visualization of Results

- Example of Noisy Image:



- Example of Denoised Image:

- Example of True Image:

Conclusion

This project successfully implemented a U-Net architecture for image denoising. The model showed promising results in reducing noise and preserving image details, as evidenced by the final PSNR value and visual inspections of the denoised images. Future work could involve experimenting with different architectures, fine-tuning hyperparameters, and applying the model to different types of noise and datasets.

References

- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv preprint arXiv:1505.04597