Ideas and Initial Planning

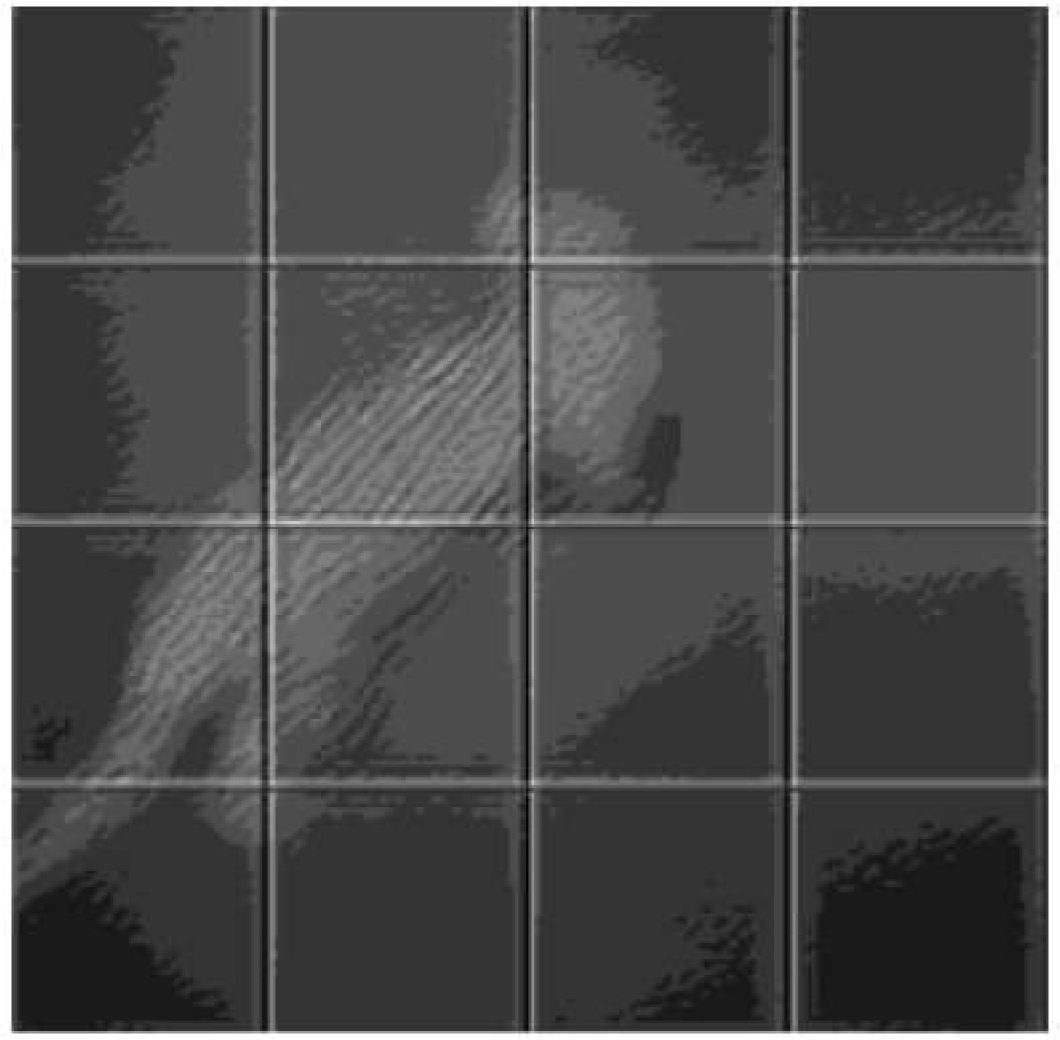
In the realm of image processing, denoising plays a critical role in enhancing image quality for applications across various domains, including medical imaging, surveillance, and digital photography. These applications often require high-fidelity images where noise can significantly degrade performance and usability. However, a key challenge lies in **blind denoising**, where the noise level and type are not known a priori, making the task more complex and less predictable.

Solution

To address the challenges of blind denoising, we developed a solution leveraging a **17-layer DnCNN** architecture for deep feature extraction. The model is applied specifically to the **Y channel** of images, as it demonstrated better results compared to using the full RGB space. By training the model on diverse noise patches, we enhanced its ability to generalize across varying noise levels and patterns. Combined with techniques like residual learning and robust preprocessing, our approach achieves effective noise reduction while retaining essential image detail

Dependencies

TensorFlow/Keras : For building and training the DnCNN model.
NumPy : Used for numerical computations and data processing.
OpenCV (cv2) : Handles image loading, resizing, and conversions.
Pillow (PIL) : Performs image manipulation and compression tasks.
os and io : For file handling and in-memory operations.
random : Used for data augmentation and random sampling.
scikit-image : Computes SSIM and other image quality metrics.
scikit-learn : Calculates MAE and other statistical metrics.
TensorBoard : Visualizes model training and performance metrics.
matplotlib (Optional) : Used for plotting graphs and visualizations.



Methodology and results

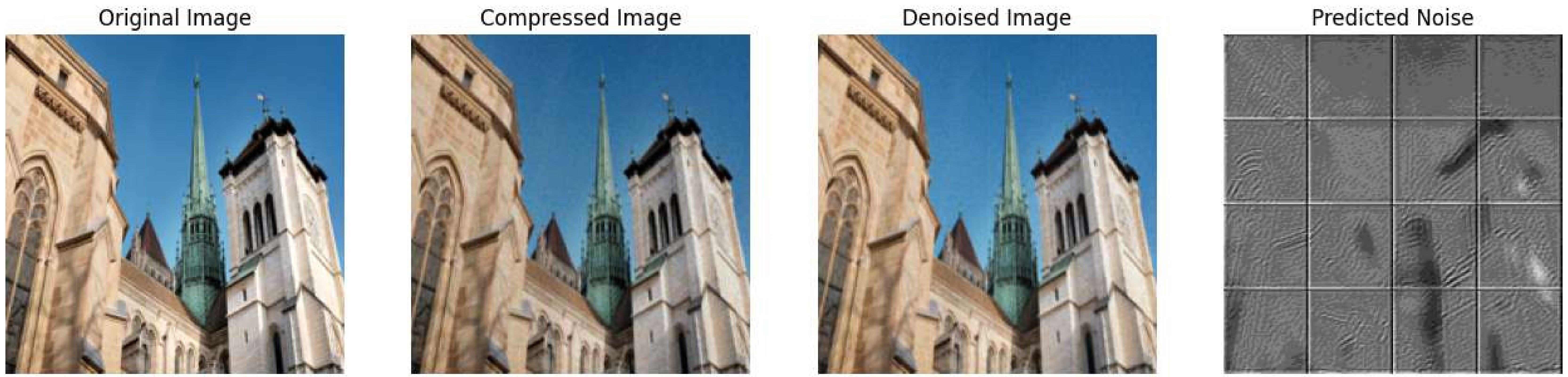


Image: 0756.png

PSNR (Original vs Denoised): 30.03 dB

PSNR (Original vs Compressed): 27.78 dB

SSIM (Original vs Denoised): 0.9088

SSIM (Original vs Compressed): 0.9041

MSE (Original vs Denoised): 0.002460

MSE (Original vs Compressed): 0.006468

MAE (Original vs Denoised): 0.037604

MAE (Original vs Compressed): 0.071046

Methodology and results

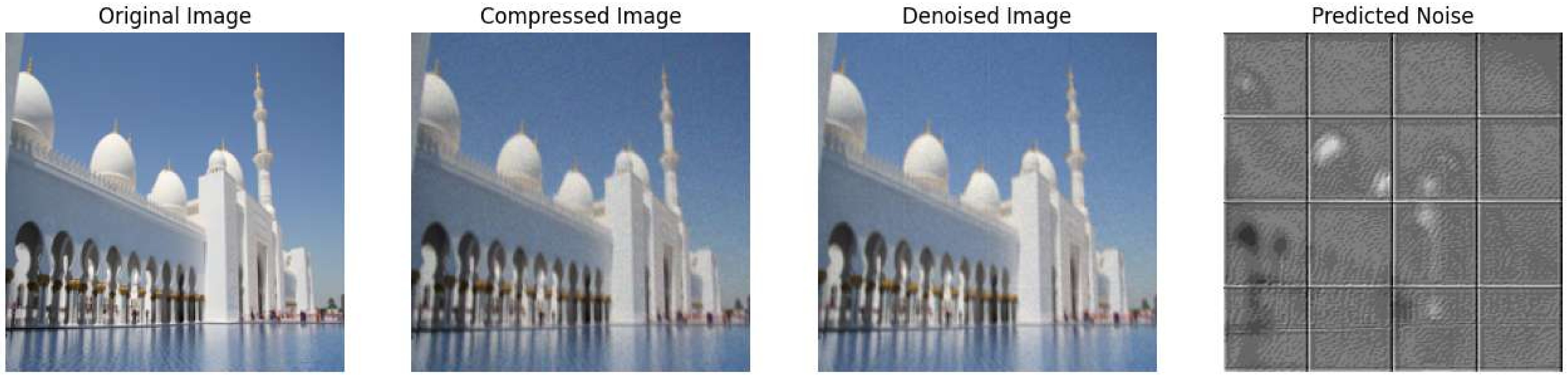


Image: 0551.png
PSNR (Original vs Denoised): 31.94 dB
PSNR (Original vs Compressed): 27.86 dB
SSIM (Original vs Denoised): 0.8321
SSIM (Original vs Compressed): 0.8264
MSE (Original vs Denoised): 0.002656
MSE (Original vs Compressed): 0.005258
MAE (Original vs Denoised): 0.030158
MAE (Original vs Compressed): 0.059567

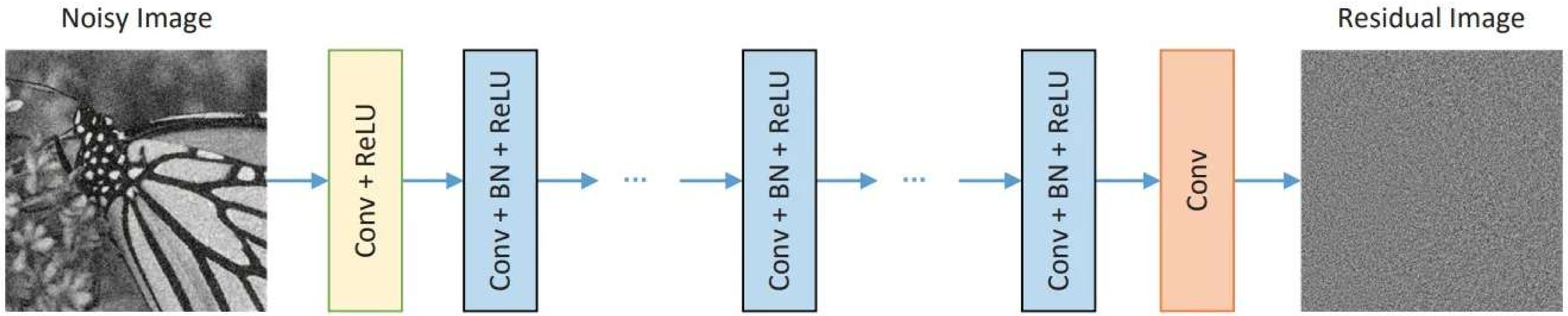
DnCNN Architecture

Summary of Architecture

- 17-layer Structure: The model uses 17 convolutional layers for deep feature extraction.
- Batch Normalization: Ensures training stability and faster convergence.
- Residual Learning: Predicts the noise map instead of the denoised image for simplicity.
- Input and Output: Processes 50×50 patches to learn noise patterns effectively.
- Y Channel Focus: Operates on the Y channel for better denoising performance than RGB.
- Activation Functions: Uses ReLU activation in all layers except the last one.
- Regularization: Includes L2 regularization and early stopping to prevent overfitting.
- Blind Denoising: Generalizes across unknown noise types without prior knowledge

Applications

- Medical imaging (e.g., denoising X-ray or MRI images)
- Photography (e.g., noise reduction in low-light images)
- Video processing (e.g., removing noise from video frames)



Results

Average PSNR (Original vs Denoised): 29.3613 dB
Average PSNR (Original vs Compressed): 29.2073 dB
Average SSIM (Original vs Denoised): 0.7565
Average SSIM (Original vs Compressed): 0.7224
Average MSE (Original vs Denoised): 0.007282
Average MSE (Original vs Compressed): 0.008953
Average MAE (Original vs Denoised): 0.058712
Average MAE (Original vs Compressed): 0.066286

Deep Das : 2201AI55
Mridul : 2201AI51
Shubhanshi : 2201AI53