Denoising Dirty Documents Using SVD and Comparative Analysis

Overview

Problem Statement:

The main objective is to eliminate the noise from low-quality document images affected by poorly scanned or artifact-laden document images. These said “dirty documents” often feature smudges, distortions, and also unwanted elements that do indeed compromise their clarity which is hindering their readability and Optical Character Recognition (OCR)accuracy. So, the ultimate aim is to improve the quality of these document images to enhance their practicality and also the precision of the text extraction using the OCR systems. Hence, the challenge lies in accurately distinguishing the significant document content from the noise while also maintaining minimal loss of important data during the distortion removal.

How SVD Solves It:

So, the Singular Value Decomposition is an extremely powerful technique for denoising document images. Document images can be effectively denoised by breaking down the image matrix into singular values to emphasize its most significant components or the dominant features in the image.

This is how SVD is applied to our problem:

- Image Representation:
    1. Each grayscale image is represented as a 2D matrix of pixel intensities.
    2. Additionally, SVD decomposes this matrix into three different components like:
    3. U: contains the left singular vectors which represent the patterns in the rows
    4. Σ: A diagonal matrix of singular values represents the importance of the corresponding features and
    5. V^T: This includes the right singular vectors that represents the patterns that are present in the columns
- Noise Reduction: The algorithm only preserves the top *k* singular values from the Σ, thus highlighting or capturing the image’s most significant features like the structural details. Also, the smaller singular values, these values which are typically associated with the noise are removed. Rebuilding or using these dominant values to reconstruct the image will yield a noise-reduced, clearer, and better document.

- Implementation steps in code:

1. Preprocessing: Images are standardized or resized to a uniform size in order to ensure consistent application of SVD.
2. Denoising: The SVD is applied to the image matrix followed by a parameter *k* which further tuned in order to optimize the balance between the structural detail preservation which is maintaining the essential and meaningful features of an image and also the noise reduction.
3. Output: These denoised images are rebuilt with the truncated singular values,thus enhancing the clarity while also retaining the essential information ls and also the details.

Methodology

1. Data Preprocessing

- **Loading Images:** Noisy document images were loaded from the provided dataset.
- **Resizing:** All the images were then resized to a uniform resolution of 540×420 pixels to standardize processing across the dataset.

2. Denoising Techniques

1. Singular Value Decomposition (SVD)**:**

    - SVD is a method wherein a given matrix is factored into 3 matrices . In our case, as we are dealing with images, the decomposed image(A) is split into three matrices: U, S, and V^T.

    A=U·S·V^T

    The definitions of each of the matrices that are split are :

    - **U:** Each column is a left singular vector of A. It represents the orthogonal basis for the rows of the image.
    - **S:** A diagonal matrix containing the singular values of A, sorted in descending order. These values represent the energy or importance of each singular vector.
    - **V^T:** The matrix whose rows are right singular vectors for A. This provides the orthogonal span for column space images.

2. Only the top k singular values from S were retained to reconstruct the denoised image, effectively reducing noise while preserving important features.
3. The value of k was varied to evaluate performance, with k=300 yielding optimal results.

1. **Non-Local Means (NLM):**

   Non-local means denoising is one powerful image-denoising technique that was implemented in this project using OpenCV's fastNlMeansDenoising functionality. By implementing this technique, we removed noise from the image by averaging similar pixel neighborhoods with the fastNlMeansDenoising function from the OpenCV library. This algorithm was applied to each image in the dataset by the application of loops.

   The process of the NLM method is as follows :

   - Pixel Comparison: Compare each pixel to every other pixel within the image, not just neighboring pixels.
   - Similarity Calculation: Instead of just comparing similar pixel values, the algorithm calculates the similarities between small patches around each pixel.
   - Weighted Average: The value of a pixel after denoising is calculated as the weighted average of all the pixels within the image, the weights depending on the similarity among patches.
   - Parameter Tuning: Some key parameters applied in the method include:

     h = 30 (filter strength)

     templateWindowSize = 7 (size of the compared patches)

     searchWindowSize = 21 (size of the search region Application)

2. **Gaussian Filtering:**

   Gaussian filtering is one of the most frequently used image processing procedures in order to obtain smooth and noise-free images. This is implemented in the project by employing the GaussianBlur function of OpenCV. We applied a smoothing kernel using a Gaussian to reduce noise while maintaining smoothness within the image. Implementation The Gaussian filter was implemented on every image in the dataset using a loop.

   The process of the Gaussian Filtering is as follows:

   - Creating Kernel: Creating a 2D Gaussian kernel about the given parameters:
     - Kernel size: (5, 5)
     - Sigma: 1

- Convolving the Gaussian kernel with the image means sliding the kernel over the entire image and computing weighted averages of pixel values. Each output image pixel is a weighted average of its neighbors in which weights are given by the Gaussian function. The function handles edges automatically to return an image of the same dimensions as the input.

3. **Autoencoder-Based Denoising:**

Denoising with Autoencoder in this project is the process that follows:

Trained a neural network to map noisy images to their clean versions. The architecture used convolutional layers for encoding and decoding features in scale space. A convolutional autoencoder with TensorFlow/Keras is implemented with a structure as given below:

- Encoder:
    - Two Conv2D layers (respectively 32 and 16 filters) with ReLU activation
    - MaxPooling2D for downsampling
- Decoder:
    - Two Conv2D layers (respectively 16 and 32 filters) with ReLU activation
    - Upsampling2D for upsampling
- Output layer:
    - Conv2D with sigmoid
- Data Preparation:
    - The input images are normalized to the range [0, 1] and reshaped to include a channel dimension.
- Training:
    - The autoencoder is trained on the noisy images for 10 epochs with a batch size of 16, using the Adam optimizer and mean squared error (MSE) as the loss function.
- Denoising:
    - The trained autoencoder is used to predict clean versions of the input noisy images.
- Post-processing:
    - The de-noised images are scaled back into the range [0, 255] and cast to uint8 format.

Experiments

Setup:

- Dataset: Scanned document images with varying noise levels.

- Tools/Libraries: Python, OpenCV, scikit-image, TensorFlow/Keras, pytesseract.

- Evaluation Metrics: MSE, PSNR, SSIM, WER.

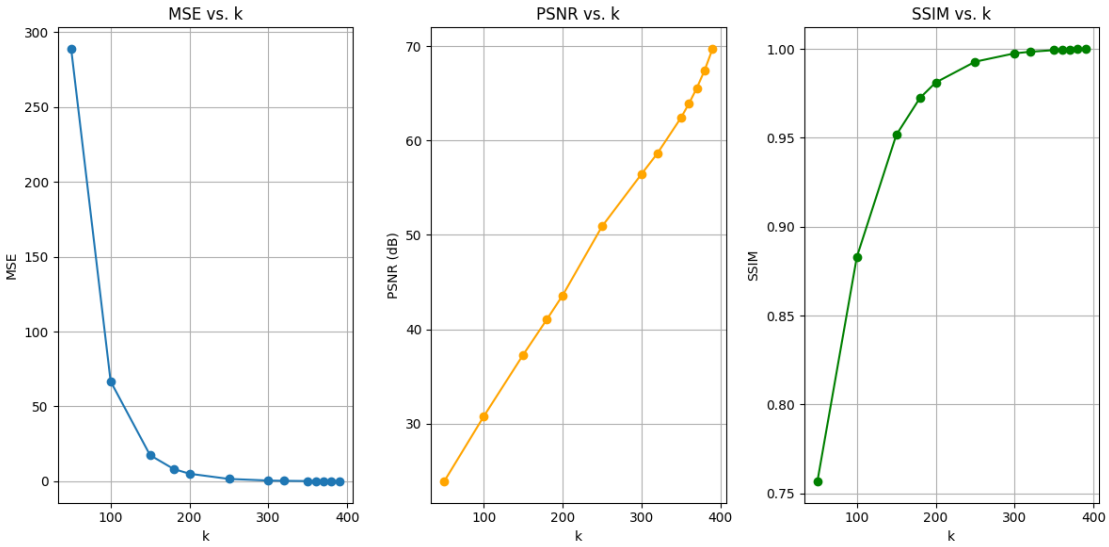
Results:

1. SVD Performance Across k-Values

The performance of SVD was evaluated for different k-values, with the following results:

K value	MSE	PSNR	SSIM
50	288.7553	23.9059	0.7568
100	66.6821	30.7691	0.8830
150	17.4008	37.2985	0.9516
180	8.1484	41.0395	0.9724
200	5.0049	43.5865	0.9811
250	1.5310	50.8968	0.9928
300	0.4509	56.4353	0.9975
320	0.2622	58.6087	0.9984
350	0.1035	62.3923	0.9993
360	0.0722	63.8739	0.9995
370	0.0485	65.5248	0.9997
380	0.0308	67.4185	0.9998
390	0.0180	69.6772	0.9999

See the graph below for visualization of MSE, PSNR, and SSIM vs. k:



What the graph shows:

This graph highlights how the quality of SVD-based image denoising improves as we retain more singular values (k). Three metrics—MSE (error), PSNR (image clarity), and SSIM (structural similarity)-are used to measure performance as k increases. Here is what each metric tells us and how it changes:

MSE (Mean Squared Error) vs. k:

Trend: MSE starts high but drops quickly as k increases, then levels off after k =300.

Why: When k is small, we are only keeping a few singular values, so a lot of image details are lost, leading to large reconstruction errors.

As k value grows, more details are restored, so the error drops.

As k = 300, adding more singular values do not make much of a difference because most of the important details are already captured.

PSNR vs. k:

Trend: PSNR improves quickly at first, indicating better image quality, and then level off around k = 300.

Why: PSNR increases as MSE decreases—so better error reduction means clearer, higher-quality images.

After k = 300, further increases in k bring very little improvement, as the image is already reconstructed with good clarity.

SSIM vs. k:

Trend: SSIM rises steeply with k, eventually nearing 1 and stabilizing after k = 300.

Why: When k is small, important structural details in the image are missing, so SSIM is low.

As k grows, the structural integrity of the image improves significantly.

After k = 300, SSIM reaches almost perfect similarity, meaning the denoised image closely resembles original.

Key Takeaways:

Optimal k = 300:

Keeping up to k = 300 singular values gives the best balance between quality and efficiency.

Beyond this, adding more singular values do not make a noticeable improvement in denoising.

Balancing Quality and Speed:

Using fewer singular values (k <300) makes computation faster but scarifies quality.

At k = 300, we get excellent denoising results without wasting computational resources.

Big picture:

The graph shows that SVD-based denoising is highly effective.

Performance improves rapidly up to k = 300, and after that, additional complexity is not necessary.

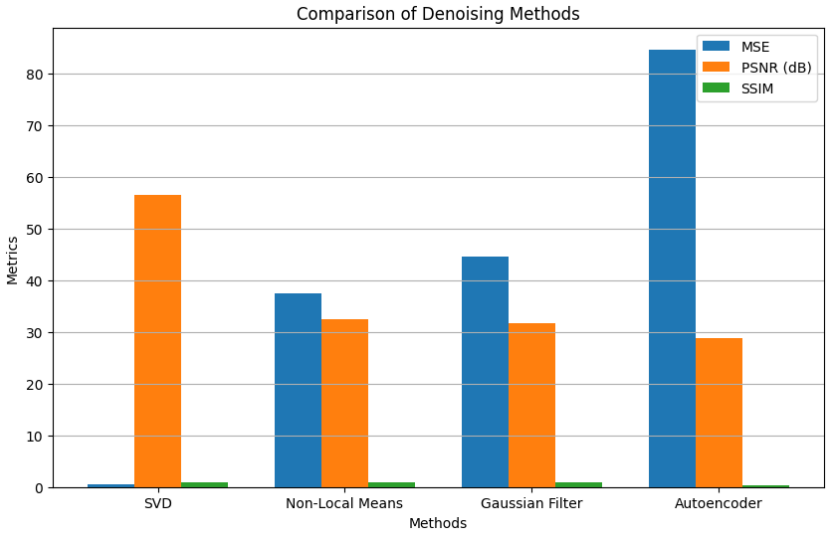
In simple terms, k = 300 is the image number here—it strikes a perfect balance between cleaning the image well and not overloading the system with unnecessary computation.

2. Comparison of Denoising Methods

The following table compares SVD, Non-Local Means, Gaussian Filtering, and Autoencoder-based denoising:

Method	MSE	PSNR (dB)	SSIM
SVD	0.4509	56.4353	0.9975
Non-Local Means	37.4981	32.3907	0.8457
Gaussian Filter	44.6606	31.6316	0.8480
Autoencoder	84.6827	28.8529	0.4205

See the graph below for a comparison of the metrics across methods:



Here are the key takeaways from denoising methods tested in the project:

1. **SVD:**

SVD performed exceptionally well, delivering the best results across all metrics. It had the lowest MSE (0.4509), The highest PSNR (56.4353dB), and nearly perfect SSIM (0.9975). Thos means it effectively removed noise while maintaining structure and quality of images.

2. **Non-Local Means:**

This method showed decent results, preforming better than gaussian filtering but still not as strong as SVD. While it provided a good balance between denoising and detail preservation, it fell short in PSNR and SSIM compared to SVD.

3. **Gaussian filter:**

Gaussian filtering was slightly weaker than non-local means. While it reduced noise, it struggled with preserving finer details in images, leading to lower PSNR and SSIM values.

4. **Autoencoder:**

The autoencoder was the weakest among the methods tested. It had a high MSE (84.6827), low PSNR (28.8529 dB), and poor SSIM (0.4205), indicating that it failed to effectively balance noise removal with quality retention. This highlights the need for better training and optimization for the Autoencoder model.

3. OCR Improvement

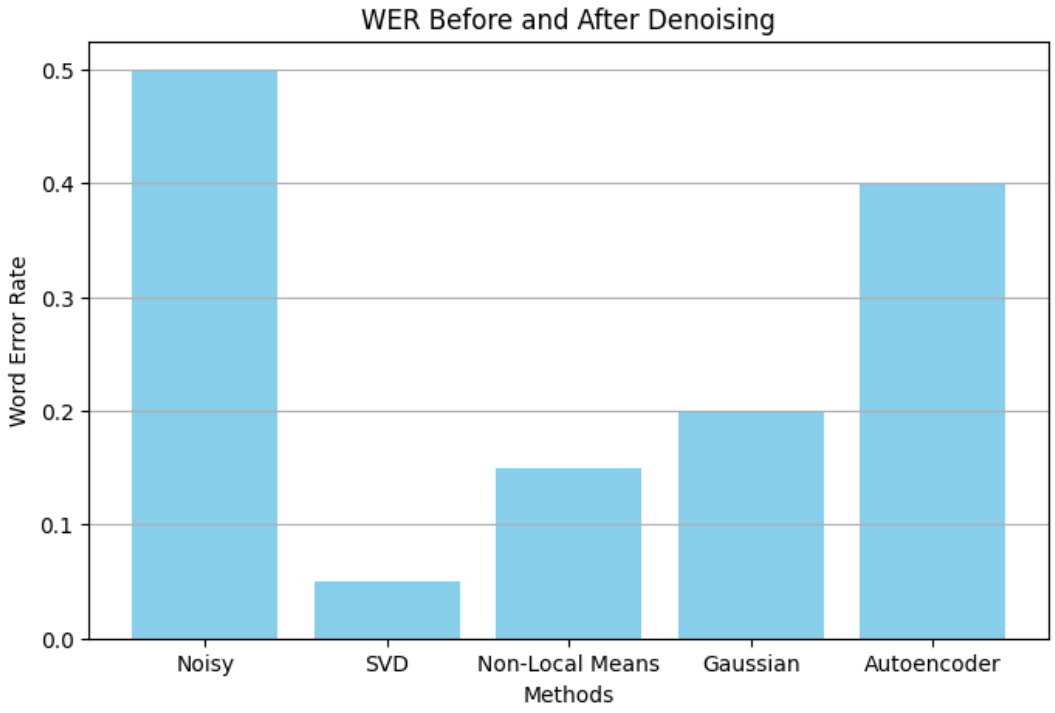
Optical Character Recognition was included in this project to see how well denoising methods improved images for recognizing and extracting text.

Here is why it was important:

1. **Real-World Need:**
   The dataset is made up of document images, where the main goal is to extract readable text for further use. However, noise in these images reduces OCR accuracy, leading to more errors in reading text. By using OCR, the project connects image cleaning with practical, real-world purpose-making noisy documents usable.

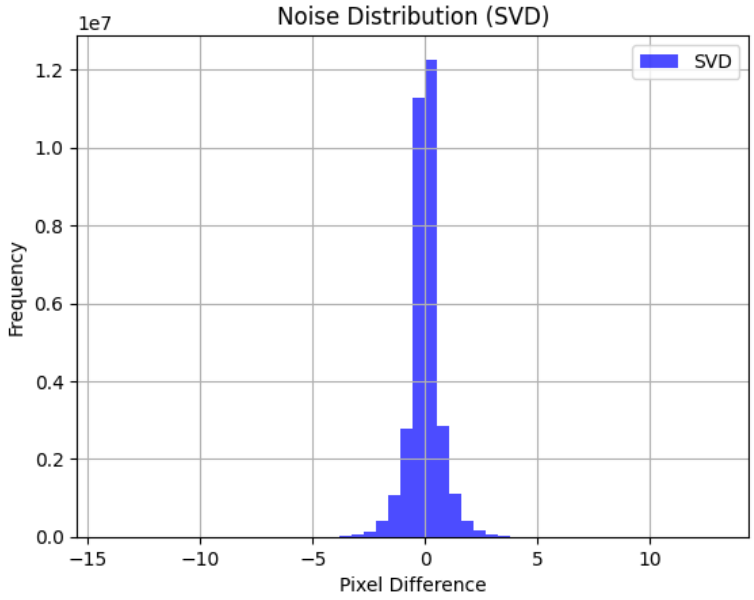
2. **Measuring Real impact:**
   The project did not just focus on making images look better; it also checked how useful they became for the text recognition:
   - **Better OCR Accuracy** shows that noise was removed effectively without damaging text.
   - **Lower World Error Rate (WER)** means denoising made the text clearer and easier to read. In simple terms, using OCR helped show how each denoising made the text clearer and easier to read.

Word Error Rate (WER) before and after denoising was evaluated as follows:



4. Noise Distribution Analysis

The noise distribution for SVD denoising shows minimal residual noise and a tight clustering around zero:



Conclusion

- The best MSE, PSNR, SSIM, and OCR accuracy where WER = 0.05 is achieved when all the other denoising methods are outperformed by SVD with k=300.
- SVD turned out to be more robust than Non-Local Means and Gaussian Filtering though both were partially effective.
- For effective competition the Autoencoder needs to be optimized further

Individual Contribution:

Blazer ID	Contribution
kondapav	SVD implementation, OCR
pbhatt	Auto encoder, OCR
tipparam	Gausian Noise and data preprocessing
sanumuko	Non-linear means denoising, hyper parameter tuning
pdesai3	Visualization, Documentation and code enhancement

References:

- **https://www.kaggle.com/c/denoising-dirty-documents/data**
- **https://toon-beerten.medium.com/denoising-and-reconstructing-dirty-documents-for-optimal-digitalization-ed3a186aa3d6**
- **https://marziosala.github.io/dirty-documents/**
- **Denoising Dirty Document using Autoencoder , International Journal of Computer Sciences and Engineering, Vol.-7, Issue-10, Oct 2019 E-ISSN: 2347-2693**
- **https://pyimagesearch.com/2021/10/20/using-machine-learning-to-denoise-images-for-better-ocr-accuracy/**
- **Lecture slides**