Assignment 1

Group members:
ShuoZhang, 530791539, szha0882
Zichong Zeng, 540075768, zzen0696
Niklas Schmitt, 540940765, nsch0022

Abstract

Machine learning is a tool for researchers to uncover the underlying patterns in datasets. Its workflow consists of five key elements: input training data, predefined hypotheses, objective functions, optimization methods, and output hypotheses [1]. When using images as input data for ML tasks, such as facial recognition and autonomous driving, image noise significantly affects model performance. This has been confirmed in numerous empirical studies (Dealing with Noise Problems in ML). Our paper addresses the problem of efficient image reconstruction from noisy data using Non-negative Matrix Factorization (NMF).

Organization: In the beginning of the report, we will briefly introduce the principles of NMF. Then, we will implement three variants of NMF to evaluate their effectiveness in image reconstruction under different noise conditions. After introducing the methods we used and analyzing the robustness of the algorithms theoretically, we will provide a detailed analysis of the experiments and the results of our study.

1 Introduction

In image reconstruction and dimension deduction problems, Non-negative Matrix Factorization (NMF) is a common matrix factorization method that decomposes a matrix into two lower-rank matrices with non-negative entries. Specifically, it decomposes a non-negative matrix $V \in R^{m \times n}$ into two lower-rank matrices $W \in R^{m \times r}$ and $H \in R^{r \times n}$, [2]such that

$$V \approx W \times H$$

Where:

- V is an $m \times n$ matrix (original matrix)
- W is an $m \times r$ matrix (basis matrix)
- H is an $r \times n$ matrix (coefficient matrix)
- r is the reduced rank or number of components

where all elements in V, W, and H are non-negative. Matrix W represents the **basis matrix**, where each column(with dimension r) can be interpreted as a latent pattern in the original data. The number r indicates the number of latent features extracted from the original data. And each row shows how much each feature from the original data is involved in forming the new latent features. Matrix H represents the **coefficient matrix**, where each column describes how the features from the basis matrix are combined to approximate different data points in the original matrix. Each row(with dimension r) describes how each implicit feature participates in the reconstruction of different data points. In the matrix multiplication $V \approx W \times H$, each element **Vij** of the original matrix **V** can be approximately reconstructed as a weighted sum of the products of elements from **W** and **H**:

$$V_{ij} \approx \sum_{k=1}^{r} W_{ik} \cdot H_{kj}$$

Thus, NMF can extract sparse features from non-negative data. Moreover, r is an important hyperparameter that determines the number of latent features in NMF [3]. When r is small, it is typically used for feature extraction tasks. In this case, fewer latent features are used to compress the data, which helps extract the most important patterns while ignoring the details. When r is a large number, more information is preserved during the matrix decomposition, which can be used for image reconstruction, as more features and details are retained, making the reconstructed result closer to the original image.

In real-world tasks, data is often accompanied by various kinds of noise, which can significantly degrade the performance of NMF. Therefore, improving the reconstruction performance and robustness of the NMF algorithm in the presence of noise is a challenging problem.

The objective of this study was to evaluate the robustness of different NMF variants in image reconstruction by comparing their performance under multiple types of noise in two different datasets. To achieve this goal, we tested several variants, including L1NMF, L2NMF, and L1NMFReg, against different types of noise, such as Gaussian noise, salt-and-pepper noise, and blocking noise. [4]. Additionally, we studied the effect of the number of decomposition components (which is r) on the robustness of NMF. Finally, we controlled the intensity of noise by adjusting the parameters of noise to test the robustness of each algorithm. To measure the performance of various NMF algorithms, we used several metrics, such as Relative Reconstruction Error (RRE), which measures the error generated during data reconstruction; Average Accuracy, which evaluates how well the predicted labels match the true labels; and Normalized Mutual Information (NMI), which assesses the similarity between the clustering results and the true classifications.

2 Related Work

Matrix decomposition technology was initially mainly used for dimension reduction and feature extraction. As research has deepened, it has been widely applied in fields such as machine learning, signal processing, and many others. . The most classical matrix decomposition methods include Singular Value Decomposition (SVD) and Principal Component Analysis (PCA), both widely used for data analysis and dimension reduction.

- **Singular Value Decomposition (SVD)**:
  - SVD is one of the earliest techniques used for matrix decomposition. It works by decomposing a matrix into three components, effectively achieving dimension deduction.While SVD performs well in many areas, one limitation is that it does not support non-negative constraints and is therefore not intuitive when working with non-negative data such as images [5] Here is the function of the SVD

$$A = U\Sigma V^T \tag{1}$$

  - **Where**
    - A is the original matrix
    - U is an $m \times m$ orthogonal matrix The columns of U are called the left singular vectors.
    - Σ is an $m \times n$ diagonal matrix.The diagonal elements are the singular values.
    - V^T is an $n \times n$ matrix . It is the transpose of an $n \times n$ orthogonal matrix. The columns of V (or rows of V^T) are called the right singular vectors.

- **Principal Component Analysis (PCA)**:
  - PCA is a linear dimension deduction method that maps high-dimensional data to a lower-dimensional space. However, PCA assumes a Gaussian distribution of the data, which means it cannot effectively handle non-negative data constraints, limiting its application to non-negative data like images . [6] Here is the function of the PCA

$$X = WZ \tag{2}$$

  - Where
    - X is an $m \times n$ matrix representing the original dataset
    - W is an $m \times k$ matrix called the **principal component matrix**, where each column is an eigenvector and represents the linear combination of features in the principal component.
    - Z is an $k \times m$ matrix representing the data projected onto the principal components, referred to as the **principal component scores**.

- **Non-negative Matrix Factorization (NMF)**:
  - The NMF (Non-negative Matrix Factorization) method includes a non-negativity constraint, making it more suitable for handling image data, where pixel values are non-negative. [7] It decomposes data into the product of two non-negative matrices, showing superior performance in image processing tasks. As a matrix decomposition method, although NMF solves the problem of non-negative constraints, it still has many shortcomings. First, When NMF is using iterative optimization methods (such as multiplication update rules or gradient descent) it may converge to a local rather than a global optimal solution, which affects the stability and reliability of the results. [2]Second, NMF is usually sensitive to the choice of initial values, and different initialization may lead to significantly different results, and this randomness may make the model lack repeatability.[8] In addition, for large-scale data, the calculation of matrix decomposition can be very time-consuming, especially if multiple iterations are required.[9] Finally, many standard matrix decomposition algorithms, such as NMF, do not perform well when dealing with sparse data because they attempt to take advantage of all non-negative values for reconstruction and may not capture sparse features effectively.[10]

3 Methods

3.1 Description of the Noise

In real-world scenarios, image data is frequently affected by noise, which can degrade the quality of both image reconstruction and denoising processes. A significant portion of this noise originates during image acquisition. Various factors contribute to this, including the inherent characteristics of the imaging sensor, such as fluctuations in illumination and sensor temperature. Additionally, the electronic circuits that interface with the sensor introduce further noise through their own electrical activity [11].

To evaluate how different NMF algorithms perform under various types of data corruption, we simulated four typical types of noise: salt-and-pepper noise, block occlusion, Laplacian noise, and Gaussian noise. For Laplacian and Gaussian noise, negative values were clipped on the positive side to meet the non-negativity condition of NMF. These types of noise were applied to the ORL and Extended YaleB dataset. Examples of these noise types, with varying degrees of application, are shown in Fig. 1 below

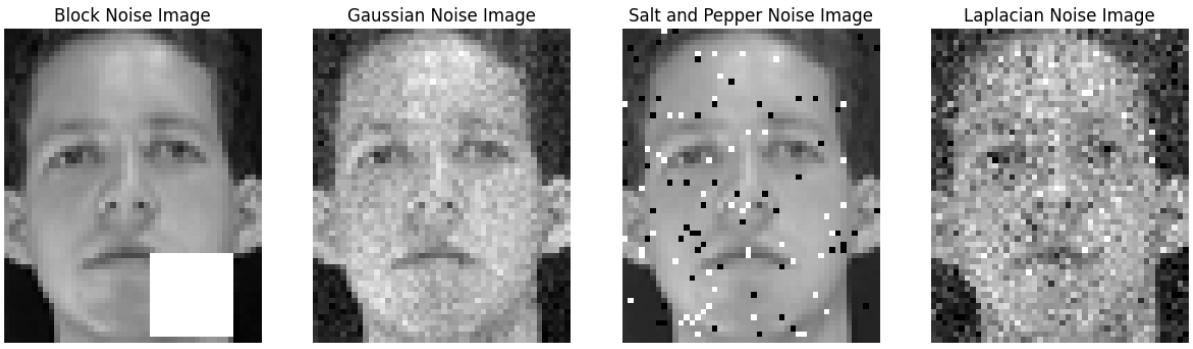

Figure 1: Examples of different types of noise applied to an image from the ORL dataset. From left to right: Block Occlusion Noise (15x15 white block occluding part of the image), Gaussian Noise (mean = 0, standard deviation = 10), Salt-and-Pepper Noise (probability = 0.05), and Laplacian Noise (location = 0, scale = 20).

3.1.1 Gaussian Noise

A common model for image noise is Gaussian noise, which is typically assumed to be additive and independent at each pixel. This type of noise is often introduced by the sensor, particularly through thermal effects such as Johnson–Nyquist noise and reset noise from capacitors [12]. In darker areas of the image, this noise, often referred to as "read noise," is primarily caused by the amplification process [12].

At higher exposures, the noise characteristics shift, and shot noise, which is signal-dependent and non-Gaussian, becomes more prominent. However, Gaussian noise remains a widely used model for noise due to its simplicity and prevalence in many imaging systems [12].

Gaussian noise can be mathematically modeled as:

$$n(x, y) = I(x, y) + \mathcal{N}(0, \sigma^2)$$

where $I(x, y)$ is the original pixel intensity at position (x, y), and $\mathcal{N}(0, \sigma^2)$ represents Gaussian noise with mean 0 and variance σ^2.

3.1.2 Laplace Noise

As described in [13], Laplace noise is generated by the Laplace distribution and shares similarities with the Gaussian distribution, though it features a sharper peak around the mean value. A comparison between the Gaussian and Laplace distributions can be seen in Figure 6, with the probability density function (PDF) of the Laplace distribution given as:

$$f(x \mid \mu, \lambda) = \frac{1}{2\lambda} \exp\left(-\frac{|x - \mu|}{\lambda}\right) \tag{3}$$

$$= \begin{cases} \frac{1}{2\lambda} \exp\left(-\frac{\mu - x}{\lambda}\right) & \text{if } x < \mu, \\ \frac{1}{2\lambda} \exp\left(-\frac{x - \mu}{\lambda}\right) & \text{if } x \geq \mu. \end{cases} \tag{4}$$

where μ represents the mean and λ is the scale parameter. As noted in [13], the PDF is expressed in terms of the absolute difference from the mean, contrasting with the Gaussian distribution, where the expression involves the squared difference. Consequently, the Laplace distribution exhibits fatter tails than the normal distribution, making it more resilient to large deviations from the mean.

As further explored in [13], the influence of the parameter λ on image noise is significant, and its impact can be observed in images extracted from the ORL dataset, as demonstrated in Figure 8.

3.1.3 Salt and Pepper Noise

Salt and Pepper (SP) noise, also referred to as impulse noise, manifests as randomly occurring black and white pixels distributed across an image, as described in [13]. This type of noise can originate from several sources, such as malfunctioning camera sensors, software or hardware failures during image capture, or transmission errors [14]. SP noise disrupts the image by introducing pixel values that range from 0 (pepper noise) to 255 (salt noise), at random locations in the image.

In typical cases, pepper noise appears with intensity values near 0, while salt noise takes values close to 255. This form of noise introduces a non-Gaussian distribution, which can pose challenges for algorithms assuming a Gaussian noise model, including standard NMF, as noted in [13]. In this work, we simulate SP noise by randomly replacing a proportion p of the total pixels in the image with either 0 or 255. The parameter p controls the percentage of corrupted pixels, and we use different values of p to evaluate the impact of noise on algorithm performance.

3.1.4 Block Occlusion

Block occlusion is a specific type of noise that significantly degrades image recognition performance by obscuring a large, contiguous region of the image. Unlike Salt-and-Pepper noise, which affects individual scattered pixels, block occlusion removes all information from a defined area, making traditional noise-reduction methods like median filtering ineffective [15].

The challenge for NMF-based methods is even greater with block occlusion, as these algorithms must rely heavily on dictionary data to reconstruct the missing section. This is particularly difficult when the occluded region is substantial, as it leads to a large portion of the image being unavailable for processing.

In our work, we simulate block occlusion by inserting a square block of size $b \times b$ into the image at a random location. This block is assigned a constant grayscale value of 255, representing a fully occluded area.

3.2 L2-Norm NMF

3.2.1 Objective and Optimization for L2 Norm

The objective for the standard NMF optimization is defined as follows:

$$\arg \min_{D \in \mathcal{D}, R \in \mathcal{R}} \|X - DR\|_F^2 \tag{5}$$

where $X \in R_+^{m \times n}$ is the non-negative data matrix we wish to approximate, $D \in R_+^{m \times k}$ is the basis matrix (dictionary), and $R \in R_+^{k \times n}$ is the coefficient matrix. Here, $\| \cdot \|_F$ denotes the Frobenius norm, which is defined as:

$$\|A\|_F = \sqrt{\sum_{i,j} |A_{ij}|^2} \tag{6}$$

This optimization problem cannot be solved analytically [13], so we employ an iterative method called the Multiplicative Update Rule (MUR) to minimize the objective function [16].

One of the unfortunate drawbacks of standard NMF is its sensitivity to outliers and noise [17]. This is due to the fact that small changes in the data matrix X can lead to large variations in the factorization results, making the method less robust.

The standard update rules for D and R are as follows:

$$D \leftarrow D \circ \frac{XR^T}{DRR^T} \tag{7}$$

$$R \leftarrow R \circ \frac{D^T X}{D^T DR} \tag{8}$$

where $\circ$ denotes element-wise multiplication, and the division is also element-wise.

In our implementation, both D and R are initialized randomly. The algorithm iteratively updates these matrices using the multiplicative rules until convergence. However, because NMF is prone to getting stuck in local minima, different initializations may lead to different results.

3.2.2 Robust analysis

Gaussian Noise Gaussian noise is a kind of noise conforming to normal distribution, which is characterized by small most noise values and less extreme noise values. This noise is randomly distributed across the data points of the image. When dealing with Gaussian noise, the goal of the L2-Norm NMF model is to minimize the reconstruction error between the original matrix X and the decomposed matrix DR. The L2 norm measures the

difference by calculating the square of the error, specifically, it sums the square of the error. Because the squared operation amplifies large error values, L2 norm is particularly sensitive to large error points (such as extreme noise) during optimization.[18]

Laplace Noise Compared with Gaussian noise, Laplacian noise has a sharper central peak, indicating that most noise values are concentrated around the mean. However, its longer tails mean that extreme noise values, though less frequent, occur more often than in Gaussian noise. As a result, Laplacian noise is more likely to produce larger deviations and outliers. The L2 norm squares the error values, which makes it particularly sensitive to large errors or extreme values. This causes the model to focus excessively on these extreme noise points during optimization, potentially overlooking the overall data distribution. Therefore, we believe that the L2-norm NMF is not very robust to Laplacian noise.

Salt and Pepper Noise and Block Occlusion Salt noise is a random occurrence of extreme noise that causes some pixel values in an image to become extreme black (0) or white (1), while most pixels remain unchanged. This noise is characterized by a small number of pixels that are disturbed to a great extent, similar to extreme outliers. Block noise means that large areas in the image are affected by noise, resulting in drastic changes in a large range of pixel values. This noise tends to appear as block areas where pixel values are skewed within the block, while pixels outside the block remain unaffected. Since L2-norm squares up the error, it produces a strong response to extreme noise points generated by salt and pepper noise and block noise. As a result, the model will focus too much on these few extreme values during optimization and ignores the major feature of images. Therefore, we believe that the robustness of L2-norm on salt-and-pepper noise and block noise is poor.

3.3 L1-Norm NMF

3.3.1 Objective function

The objective function of L1 norm NMF is defined as follow[19]:

$$C(D, R) = \frac{1}{2}\|X - DR\|_F^2 + \lambda\|R\|_1 \tag{9}$$

The first part of the objective function represents the reconstruction error, and the second term corresponds to the $L1$-norm on matrix R, encouraging sparsity by promoting more zero elements in matrix R. According to this formulation, only matrix R requires regularization since the matrix D serves as a dictionary matrix which acting as a set of basis vectors.

3.3.2 Optimization

The equation of L1 norm is defined as follow:

$$\lambda\|R\|_1 = \lambda\sum_{i,j}|R_{ij}| \tag{10}$$

According to the L1 norm definition, we can define the equation of optimization as follow:

$$\min_{D,R}\frac{1}{2}\|X - DR\|^2 + \lambda\sum_{i,j}|R_{ij}| \tag{11}$$

In this section, the goal is to minimize the objective function. Methods such as Multiplicative Update Rules (MUR) or gradient descent are commonly applied to the update rules. This part introduced the update rule based on the MUR approach. The update rule for D and R are defined as follow:

$$D \leftarrow D \times \frac{XR^T}{DRR^T} \tag{12}$$

$$R \leftarrow R \times \frac{D^T X}{D^T DR + \lambda} \tag{13}$$

While using this update rule, the objective of 9 may reach the global minimum [19].

3.3.3 Robust analysis

Gaussian Noise L1-norm imposes linear penalties on all error values without the effect of amplifying larger error points. Since the extreme value of Gaussian noise is less, L2-norm is better at reducing the influence of these larger noise points through square punishment, thus achieving better noise suppression effect. However, L1-norm cannot make full use of noise distribution characteristics for optimization. Therefore, we believe that L1-norm NMF is relatively poor in robustness to Gaussian noise. L2-norm is more suitable for dealing with Gaussian noise

Laplace Noise L1-norm NMF is more robust to Laplacian noise. Since L1-norm only calculates absolute values for errors, it does not over-amplify the effect of extreme values as L2-norm does. Especially in the case that Laplacian noise is easy to produce extreme values, L1-norm can effectively prevent the model from overfitting these extreme noise points, and the overall performance is better.

Salt and Pepper Noise and Block Occlusion When dealing with salt-and-pepper noise and block noise, because L1-norm has a mild penalty for extreme errors, it does not pay too much attention to these few pixels affected by noise, but can process the overall data more stably. L1-norm can better ignore these extreme noise points, thereby avoiding overfitting noise.

3.4 L1-Norm Regularization NMF

3.4.1 Objective function

The core of the objective function is to minimize the reconstruction error. But unlike the other method in this report , L1- regularization is introduced to control the sparsity of the matrix. Here is the math explanation of the method.

$$\min_{D,R} \|X - DR\|_1 + \lambda(\|D\|_1 + \|R\|_1)$$

- Where
- X:The input data matrix (original matrix).
- D:The basis matrix which contains the learned components that describe the data.
- R:The coefficient matrix which provides the weights for combining the components in D.
- $\|X - DR\|_1$:The L1 norm of the reconstruction error
- $\lambda(\|D\|_1 + \|R\|_1)$The L1 regularization term controls the sparsity of the basis matrix D and coefficient matrix R

This objective function is implemented in the class's objective function and consists of two parts:

reconstruction-error: reconstruction error between matrix X and D regularization :L1 regularization term, by regularization parameters controls sparsity.

3.4.2 Optimization

To optimize the loss function we have many options to choose. One of the most popular method is Gradient descent(GD). It is a flexible, efficcient and scalable optimization technique. GD can work in many different objective function and support Regularization. And it is easy to understand and interpretable. In our optimization problem, Gradient Descent(GD) faces some unique challenges due to the non-differentiability of the objective function.**L1-norm** is defined as the sum of the absolute values of the coefficients:

$$\|v\|_1 = \sum_i |v_i|$$

The issue is that the gradient of the absolute value function is undefined at zero. Gradient descent relies on well-defined gradients to make small, incremental updates to the parameters. When any parameter vi is close to zero , the gradient is not well-defined, making standard gradient descent difficult to apply.

To overcome the issues with gradient descent and L1-norm regularization, **proximal gradient descent** is typically used. The core idea is that after each update, the proximal operator is applied to the variables to ensure they remain within the constrained range. The common proximal operator is the soft threshold operation.Here is the function:

$$Prox_\lambda(v) = \text{sign}(v) \cdot \max(|v| - \lambda, 0)$$

And because of non-negative nature, the function is simplified as:

$$Prox_\lambda(v) = \max(|v| - \lambda, 0)$$

After the presentation of the optimization method selection,,we introduce the specific optimization process:

1. Initialization
   (a) D and R are initialized randomly with non-negative values.
2. Gradient Descent Updates:
   (a) Gradient Update for R
      i. The gradient with respect to R is calculated as :

$$\nabla_R = -\mathbf{D}^T \cdot \text{sign}(X - \mathbf{D}R)$$

ii. This gradient is used to update R through a gradient descent step

$$R_{\text{temp}} = R - \eta \cdot \nabla_R$$

(b) Proximal Operator for R

i. After the gradient step, the proximal operator for **L1 regularization** (soft thresholding) is applied:

$$R = \max(0, R_{\text{temp}} - \eta \cdot \lambda_{\text{reg}})$$

(c) Gradient Update for D

i. The gradient with respect to D is calculated as :

$$\nabla_D = -\text{sign}(X - \mathbf{D}R) \cdot \mathbf{R}^T$$

ii. The same procedure as for `R` is applied, with a gradient step:

$$D_{\text{temp}} = D - \eta \cdot \nabla_D$$

(d) Proximal Operator for D

$$D = \max(0, D_{\text{temp}} - \eta \cdot \lambda_{\text{reg}})$$

3. Stopping Criterion

We check the value of objective function in each iteration. If the objective value falls below the specified tolerance the algorithm will converges and stops early. Otherwise, it continues updating for a maximum of iterations.

$$\text{tol} = \|X - DR\|_1 + \lambda \left(\|D\|_1 + \|R\|_1\right)$$

3.4.3 Robust analysis

Gaussian Noise By introducing sparsity, L1 regularization may compress some feature weights to zero, resulting in reduced ability to capture subtle features. For Gaussian noise, this sparsity has no obvious advantage because Gaussian noise does not have a large number of extreme values. Models may miss some normal subtle fluctuations, leading to over-simplification. Therefore, L1-Norm with regularization will performs poorly when dealing with Gaussian noise

Laplace Noise Since L1 regularization tends to compress the partial weight to zero, it can effectively deal with the long tail characteristics of Laplace noise. Extreme noise points are not over-amplified by L1-norm, but are appropriately punished, so this algorithm can balance the long-tail distribution of noise well. And the sparsity caused by regularization makes the model more focused on important features and less sensitive to extreme values in Laplacian noise, which helps to avoid overfitting the model due to these extreme values.

Salt and Pepper Noise L1-norm with regularization is the most robust algorithm to salt-and-pepper noise and performance. Because this noise affects only a small number of pixels, L1 regularization can effectively ignore these extreme values by sparring the model, thereby reducing the model's excessive attention to a small number of anomalies.

Block Occlusion Blocky noise can introduce a large number of error points in a local area. The sparse performance of L1 regularization can balance the adaptability of the model to local noise and prevent overfitting of large noise regions

4 Experiment

This section highlights the key findings identified in our analysis of the various algorithms and tries to explain them. It includes an overview of the intensity of noise and the dataset will be presented, along with insights into the inner algorithmic design decisions of the different NMF variants. Additionally, it provides details on the experimental setup employed in the study.

4.1 Datasets

In the experiment, we used two face image databases: the ORL Face Database [20] and the Yale B Database (Cropped) [21]. The ORL database contains 400 images from 40 individuals, with variations in lighting, facial expressions, and details like glasses. All subjects are in an upright, frontal position against a dark background, with each image sized at 92x112 pixels and 256 gray-scale levels. The Extended Yale B database, comprising 2,414 frontal-face images from 38 individuals, provides approximately 64 images per subject, captured under different lighting conditions and varying facial expressions, with each image sized at 192x168 pixels.

4.2 Testing Procedure

We conducted three tests on the algorithms using consistent parameters, specifically focusing on the strength of the applied noise and the number of components. We utilized 100% of the available data and reduced the image quality by a factor of three to conserve computational resources. The various noise configurations and the number of components tested are as follows:

- **Rank k:** The number of components was varied from 10 to 50 in increments of 10.
- **Block Occlusion:** Block sizes were tested from 5 to 15 in steps of 5.
- **Gaussian Noise:** The mean was set to 0, and the standard deviation ranged from 10 to 30 in steps of 10.
- **Laplacian Noise:** The loc was set to 0, and the scale ranged from 10 to 30 in steps of 10.
- **Salt & Pepper Noise:** Probability values ranged from 0.05% to 0.15% in increments of 0.05%.

To ensure a fair comparison, we applied the same stopping criteria across all algorithms. Each algorithm was allowed a maximum of 100 iterations to iteratively improve the objective function, with a convergence threshold set to 1×10^{-4}. This threshold served as an additional stopping criterion if the improvement between iterations fell below this value.

The dictionary matrices D and R were initialized randomly, with values uniformly sampled from the interval $[0, 1]$.

The hyperparameters for the L1-norm regularization were set as follows: $\lambda = 50$ and $\eta = 0.01$. λ is the regularization parameter in L1-norm regularization. Setting λ to 50 indicates that we have a strong requirement for sparsity, that is, we want more values in the dictionary matrix D and R to approach zero. This is also how to distinguish it from the L1-norm NMF algorithm. η is the learning rate associated with the optimization algorithm that controls the step size of the gradient update. Set η =0.01 to ensure stability during optimization and to avoid large jumps that cause failure to converge. This was preliminarily determined by experiments

4.3 Evaluation Metrics

In this experiment, we evaluated the performance and robustness of the different NMF algorithms using three metrics: (1) Relative Reconstruction Errors (RRE), (2) Average Accuracy (Acc), and (3) Normalized Mutual Information (NMI).

- **Relative Reconstruction Errors:** To assess the quality of reconstruction, let V denote the noisy dataset and $\hat{V}$ denote the clean dataset. With W and H representing the factorization results on V, the relative reconstruction error is calculated as:

$$RRE = \frac{\|\hat{V} - WH\|_F}{\|\hat{V}\|_F}$$

  where $\|\cdot\|_F$ denotes the Frobenius norm.

- **Average Accuracy:** After factorization of V into W and H, clustering is performed using algorithms such as K-means (applied in this study), with the number of clusters equal to the number of classes. The accuracy of the predicted labels, Y_{pred}, is then computed as:

$$\text{Acc}(Y, Y_{\text{pred}}) = \frac{1}{n} \sum_{i=1}^{n} 1\{Y_{\text{pred}}(i) == Y(i)\}$$

  where Y is the true label, and $1\{\cdot\}$ is the indicator function.

- **Normalized Mutual Information:** To evaluate the mutual information between the true and predicted labels, Y and Y_{pred}, the NMI is calculated as:

$$\text{NMI}(Y, Y_{\text{pred}}) = \frac{2I(Y, Y_{\text{pred}})}{H(Y) + H(Y_{\text{pred}})}$$

  where $I(\cdot, \cdot)$ represents mutual information and $H(\cdot)$ denotes entropy.

4.4 Result evaluation

The following plots are based on our tests conducted on the ORL dataset, utilizing the lowest noise levels for each noise type.

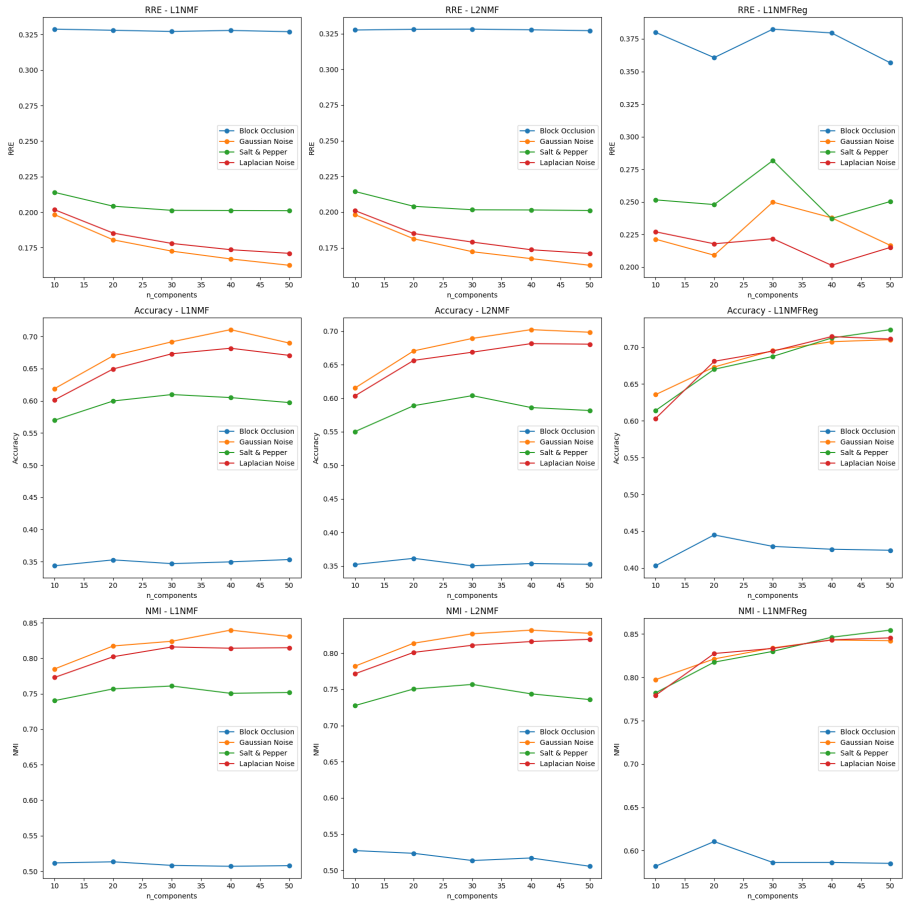

Figure 2: This figure highlights the influence of different types of noise on the performance of the algorithms with respect to the number of components. The tests were conducted on the ORL dataset, using the lowest noise levels for each noise type.

It is evident from the results that all analyzed algorithms struggle the most with denoising images affected by block occlusion noise. The Relative Reconstruction Error (RRE) is up to 50% higher compared to other noise types, and this trend is consistent across the other evaluation metrics. Interestingly, the number of components does not significantly impact the performance for algorithms such as L1NMF and L2NMF in the presence of block occlusion noise.

The difficulty in denoising block occlusion noise arises due to the structured nature of the noise, which covers large, contiguous regions of the image, effectively removing significant portions of visual information. This type of noise is particularly challenging for NMF methods, which must rely more heavily on dictionary information to compensate for the large areas of missing data. NMF methods generally assume the noise to be Gaussian in nature, as noted by Lee and Seung [7], which aligns with our experimental results. Gaussian noise consistently yielded the best results, followed by Laplacian noise, which, due to its similar characteristics to Gaussian noise, performed better than other types of noise tested.

Increasing the number of components does not improve performance with block occlusion noise because large sections of the image are missing, limiting NMF's ability to recover the fine features. More components simply add redundancy without effectively reconstructing the occluded areas.

In general, increasing the number of components improves results for other types of noise, as it allows the model to capture finer details and better represent the underlying data. However, in majority of cases, the results plateau or even degrade due to overfitting, where too many components start modeling noise or irrelevant features instead of the true structure. This happens because the feature space becomes overly dense, and the sparsity assumption, crucial for NMF's effectiveness, is violated, leading to diminished performance.

The flowing plot shows the result of our tests on the Extended YaleB dataset.

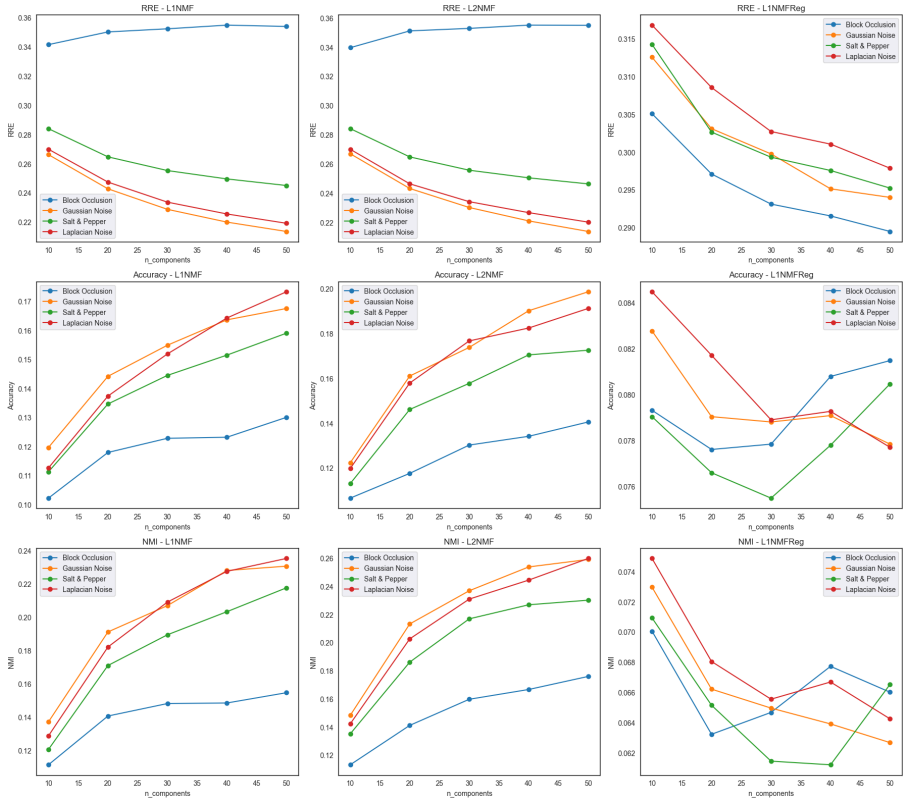

Figure 3: This figure highlights the influence of different types of noise on the performance of the algorithms with respect to the number of components. The tests were conducted on the Extended YaleB dataset dataset, using the lowest noise levels for each noise type.

Overall, three algorithms perform worse in YaleB dataset than ORL dataset. The accuracy score and NMI value of those three methods decrease a lot. Since the images in the YaleB dataset have less information and improve the difficulty to reconstruct the image. Block occlusion noise still affect the reconstruct performance most compare to other noise. Like the performance in ORL dataset, the accuracy score and NMI value of L1-norm NMF and L2-norm NMF increased as the number of components increase. However, those value of L1 regulization NMF declined as the number of components increase.

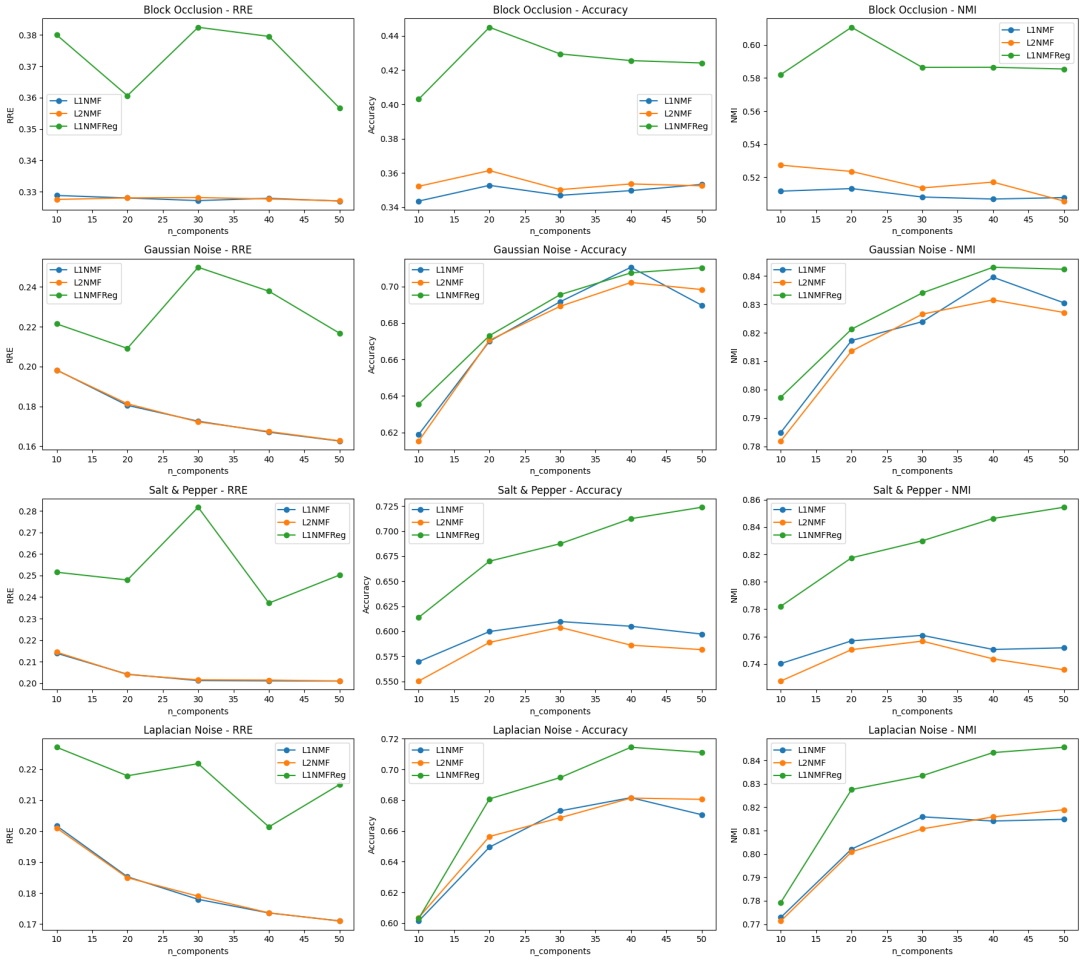

Figure 4: Comparison of NMF algorithms under different noise conditions

In the figure, each row represents a specific type of noise, while the evaluation metric remains consistent across the columns. The Y-axis shows the metric score, and the X-axis represents the increasing number of components used in the factorization. The graphs highlight the performance differences between various NMF variants, particularly with respect to Relative Reconstruction Error (RRE), Accuracy, and Normalized Mutual Information (NMI).

As the number of components increases, the RRE value of L1reg is higher than other two method, the value of L1-NMF and L2- NMF is lower and more stable than L1 regulization. The average accuracy score of L1reg is higher than other two method and it increased as the number of components rise. All NMI value of three NMF methods declined as the number of components rise but L1 regulization is better than other two methods. Therefore, L1 reg perform better when facing block occlusion noise.

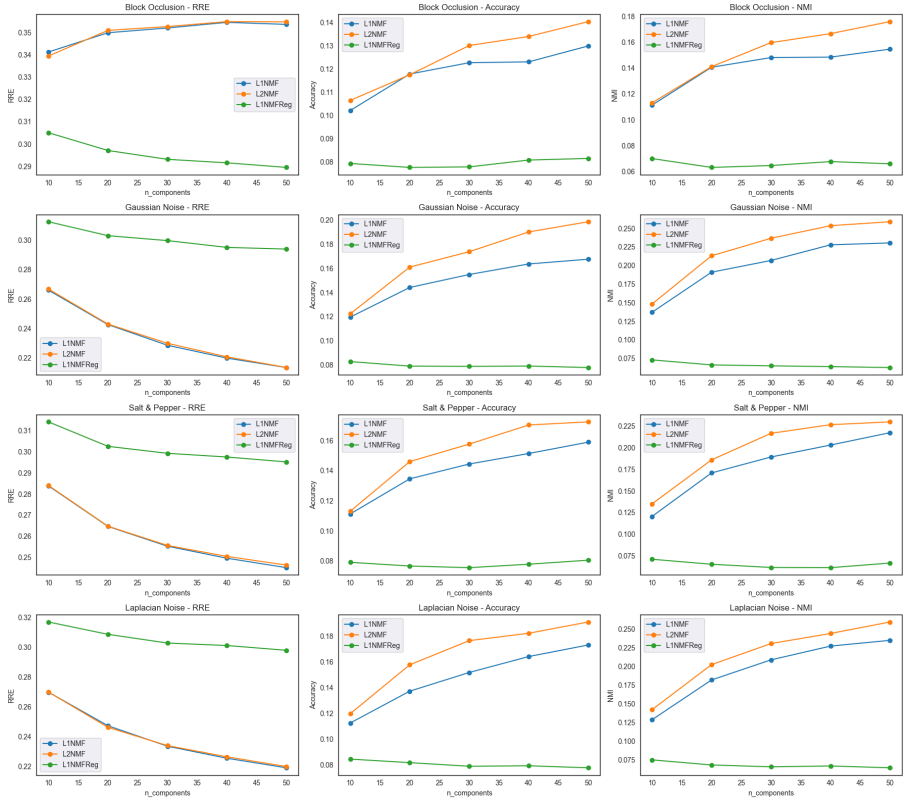

Figure 5: Comparison of NMF algorithms under different noise conditions on YaleB dataset

After changing the dataset from ORL to YaleB, all three methods performs worse.The value of RRE is similar but the accuracy score and NMI value decrease a lot. L1 regulization become the worest methods among three algorithms.

4.5 Relative Reconstruction Error (RRE)

For RRE, the differences between L1-NMF and L2-NMF are relatively small and not statistically significant. This can be attributed to the fact that RRE, which measures the difference between the reconstructed and original data, tends to favor models that minimize overall reconstruction error. Both L1 and L2 norms are effective at reducing this error, and without significant outliers, the L1 regularization (which is more robust to outliers) does not provide a notable advantage in this context. This suggests that the dataset does not contain a large number of outliers, which L1-NMF would typically handle more effectively.

Interestingly, the RRE is higher for L1-NMF with regularization compared to its unregularized counterpart. This is likely due to the regularization term penalizing certain model parameters, which may lead to less flexible representations that prioritize sparsity or robustness over minimizing the overall reconstruction error. The L1 regularization, by encouraging sparsity, can reduce the model's capacity to fully capture the finer details of the data, particularly in cases where the primary objective is accurate reconstruction rather than the identification of sparse or key features.

4.6 Accuracy and Normalized Mutual Information (NMI)

In contrast, L1-NMF with regularization outperforms the other methods on the Accuracy and NMI metrics, particularly for noise types other than Gaussian. This result is not surprising, as the canonical form of NMF assumes Gaussian noise, leading to optimized performance for Gaussian-contaminated data. However, for noise types such as Salt & Pepper or Laplacian noise, L1-NMF with regularization excels because of its ability to handle non-Gaussian noise more effectively. The regularization promotes a more structured factorization, which helps to capture the latent relationships between features, improving clustering performance (as reflected in the Accuracy and NMI metrics).

The superior performance of L1-NMF with regularization on NMI and Accuracy metrics, compared to RRE, can be explained by the nature of these metrics. While RRE focuses on pixel-wise reconstruction, NMI and Accuracy are more sensitive to the global structure and feature clustering in the data. The L1 regularization encourages a sparse and robust representation, which is less effective in pixel-level reconstruction (hence higher RRE) but

more useful for capturing essential patterns that contribute to improved clustering and classification outcomes. Therefore, L1-NMF with regularization shows an advantage in metrics that prioritize high-level feature extraction over raw reconstruction accuracy.

4.7 Analysis of Standard Deviations

The following table displays the standard deviation table of evaluation metrics value.

Metrics / Components	n=10	n=20	n=30	n=40	n=50
RRE	L1NMF: 0.001242	L1NMF: 0.000783	L1NMF: 0.000444	L1NMF: 0.000514	L1NMF: 0.000442
-	L2NMF: 0.001388	L2NMF: 0.000861	L2NMF: 0.000507	L2NMF: 0.000611	L2NMF: 0.000526
-	L1NMFReg: 0.016474	L1NMFReg: 0.017736	L1NMFReg: 0.014030	L1NMFReg: 0.023447	L1NMFReg: 0.014155
ACC	L1NMF: 0.014912	L1NMF: 0.011967	L1NMF: 0.012551	L1NMF: 0.009205	L1NMF: 0.013619
-	L2NMF: 0.014922	L2NMF: 0.016415	L2NMF: 0.015052	L2NMF: 0.017602	L2NMF: 0.009097
-	L1NMFReg: 0.016610	L1NMFReg: 0.014280	L1NMFReg: 0.015505	L1NMFReg: 0.020342	L1NMFReg: 0.016306
NMI	L1NMF: 0.008982	L1NMF: 0.008647	L1NMF: 0.008712	L1NMF: 0.006064	L1NMF: 0.009755
-	L2NMF: 0.010258	L2NMF: 0.011449	L2NMF: 0.009501	L2NMF: 0.010411	L2NMF: 0.006423
-	L1NMFReg: 0.012548	L1NMFReg: 0.009253	L1NMFReg: 0.010466	L1NMFReg: 0.013644	L1NMFReg: 0.009634

Table 1: Standard Deviation Table

Table 1 presents the standard deviations of three performance metrics—Relative Reconstruction Error (RRE), Clustering Accuracy (ACC), and Normalized Mutual Information (NMI)—for three Non-negative Matrix Factorization (NMF) algorithms (L1NMF, L2NMF, and L1NMFReg) across varying component numbers ($n = 10, 20, 30, 40, 50$).

The standard deviations for L1NMF and L2NMF are consistently low across all metrics and component numbers, with RRE values ranging approximately from 0.0004 to 0.0014. This indicates stable and robust performance of these algorithms. In contrast, L1NMFReg exhibits significantly higher standard deviations, particularly in the RRE metric, where values range from approximately 0.0140 to 0.0234. This suggests greater variability in reconstruction performance for L1NMFReg, potentially due to the inclusion of a regularization term that introduces additional complexity and sensitivity to parameter settings.

For the ACC and NMI metrics, all algorithms display relatively low and comparable standard deviations, generally between 0.0060 and 0.0203. Although L1NMFReg sometimes has slightly higher values, the differences are less pronounced compared to those observed in RRE. This indicates that the variability introduced by the regularization term affects reconstruction error more significantly than clustering performance metrics.

The higher standard deviations observed in L1NMFReg may be attributed to the challenges associated with regularization in NMF algorithms. Regularization terms, while beneficial for promoting sparsity or incorporating prior knowledge, can complicate the optimization landscape and make algorithms more sensitive to initialization and parameter tuning [16]. This sensitivity can lead to greater fluctuations in performance across different runs, resulting in higher standard deviations.

4.8 Reflection

Each algorithm has its advantages and disadvantages in dealing with different noises. The regularized L1NMF shows a large advantage in the face of non-Gaussian noise, similarly, it has no advantage in the face of Gaussian noise. Although the introduction of regularization improves the robustness of the model to sparse data and outliers, it also limits the reconstruction ability of the model in some cases. In addition, the analysis of standard deviation allowed me to further understand the importance of algorithm stability and how to enhance the performance of the model through parameter optimization.

5 Conclusion

Based on the experimental results, we can conclude that each algorithm we implemented performs better on small datasets and worse on larger datasets. However, L2-norm and L1-norm NMF outperform L1-norm with regularization on small datasets. This is because applying large penalty terms to small datasets results in significant information loss, leading to poorer performance of L1-norm with regularization. Conversely, on larger datasets, the regularization parameter proves effective, which is why L1-norm with regularization consistently performs the best on larger datasets.

Some of the theoretical analyses are also verified. In the presence of Gaussian noise, L2-norm NMF outperforms L1-norm NMF. However, when the data contains many extreme noise values, such as Laplacian noise or salt-and-pepper noise, L1-norm NMF demonstrates superior performance.

In future research, we aim to further explore the robustness of the NMF algorithm by simulating a broader variety of noise types and combinations of multiple noise sources. This will include testing the algorithm under conditions of mixed noise, such as combining Gaussian noise with salt-and-pepper noise or Laplacian noise.

Another improvement direction is to enhance the preprocessing phase of the algorithm and use singular value decomposition (SVD) to initialize the matrices D and R . We hope to speed up the convergence process of the NMF algorithm, as this approach often provides a more efficient starting point for decomposition. This method may bring faster and more stable convergence, especially when dealing with complex or noisy data, and improve the overall efficiency and effect of image reconstruction.

One issue is that convergence is not easy for L1-regularized norms. Advanced optimization algorithms like Adam use momentum and filtering techniques to accelerate convergence. However, these techniques were not used in the experiments, leading to slower convergence in some cases.

References

[1] Yu Yao. Advanced machine learning (comp 5328) lecture notes. `https://www.sydney.edu.au/units/COMP5328`, 2024. Lecture notes.

[2] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[3] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[4] Naiyang Guan, Dacheng Tao, Zhigang Luo, and Bo Yuan. Non-negative matrix factorization with l2,1-norm for clustering and representation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(7):1013–1027, 2012.

[5] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420, 1970.

[6] Ian T Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002.

[7] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Proc. Int. Conf. Neural Inf. Process Syst.*, pages 556–562, 2001.

[8] H. Kim and H. Park. Non-negative matrix factorization based on alternating non-negativity constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):859–872, 2011.

[9] Andrzej Cichocki and Anh-Huy Phan. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 92(3):708–721, 2009.

[10] Nicolas Gillis. The why and how of nonnegative matrix factorization. In *Regularization, Optimization, Kernels, and Support Vector Machines*, pages 257–291. CRC Press, 2014.

[11] Philippe Cattin. Image restoration: Introduction to signal and image processing. MIAC, University of Basel, 2012. Archived from the original on 2016-09-18. Retrieved 11 October 2013.

[12] Jun Ohta. *Smart CMOS Image Sensors and Applications*. CRC Press, 2008.

[13] Alex Díaz and Damian Steele. Analysis of the robustness of nmf algorithms. *arXiv preprint arXiv:2106.02213*, 2021.

[14] A. K. Boyat and B. K. Joshi. A review paper: Noise models in digital image processing. *Signal & Image Processing: An International Journal*, 2015.

[15] Pengwei Yang, Chongyangzi Teng, and Jack George Mangos. Contaminated images recovery by implementing non-negative matrix factorisation, 2023.

[16] Christos Boutsidis and Efstratios Gallopoulos. Svd based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition*, 41(4):1350–1362, 2008.

[17] Pengwei Yang, Chongyangzi Teng, and Jack Mangos. Contaminated images recovery by implementing non-negative matrix factorisation, 11 2022.

[18] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[19] P.O. Hoyer. Non-negative sparse coding. In *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*, 2002.

[20] F.S. Samaria and A.C. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pages 138–142, 1994.

[21] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.