
Assignment 1

Tutors: Songhua Wu

Group members:

1. Tulika Srivastava | 500657898 | tshr3215@uni.sydney.edu.au
2. Pooja Vijay Mahajan | 510282930 | pmah0895@uni.sydney.edu.au

Abstract

Non-negative matrix factorization or NMF is a type of unsupervised machine learning which is used to perform pattern recognition, dimensionality reduction, signal processing, feature extraction and data reconstruction [9]. Our objective was to analyze the robustness of two NMF algorithms (Frobenius norm and L_{2,1} norm) on two real-world face datasets (ORL dataset and Extended YaleB dataset) when the data is manually corrupted by different types of noise. The report organization is as follows. In section 1 of the report, the main idea behind NMF and its applications are introduced, followed by an overview of methods used in our experiments. In section 2 of literature review, the main idea of NMF methods are reviewed, accompanied by a list of their advantages and disadvantages. In Section 3 the objective functions and optimization steps have been described. Additionally, our choices of noise and the robustness of each algorithm from a theoretical perspective have been justified. In section 4, the dataset, algorithms and noise used in the experiment setup have been described. Finally, in section 5, a summary of results has been concluded and future scope has been discussed.

1 Introduction

1.1 Problem and it's significance

In general, the data is usually corrupted when its unprocessed, due to the presence of noises and outliers [1]. Noisy data incurs more storage space, increased computational power and inconsistencies during data analysis [2]. Hence, it is often necessary to reduce the feature space of a dataset for the ease of computation without losing important information. Non-negative matrix factorization (NMF) plays a significant role here because it is capable of extracting sparseness automatically, making it convenient to visualize the basis columns [3] which are non-negative in nature.

1.2 An overview of NMF

Using the techniques of Dictionary Learning, NMF is used to reduce the dimensions of the data into a linear combination of bases [4]. If X is a non-negative matrix of dimension $(m \times n)$, NMF can be used to obtain low rank matrices D and R (sizes $m \times k$ and $k \times n$) respectively such that $X \approx DR$ [4].

1.3 NMF applications

NMF has a lot of use cases ranging from signal processing [5], image processing [6], data mining [7], clustering [8], speech processing systems [9], optical data processing [9], biomedical engineering [9], pattern recognition [10] and computational mathematics [9]. Specifically for image processing,

NMF is used to compress images such that the original input is transformed into a dense vector with smaller dimensions and comprehensive features [11]. Additionally, some NMF algorithms are also used for isolating and removing noise which can lead to the reconstruction of the original image [12].

1.4 Overview of methods

In this assignment, we have explored the variations of NMF algorithms for image processing. Based on extensive literature review, we chose to evaluate the robustness of two NMF algorithms ie. Frobenius norm and L2,1 norm. Three evaluation metrics were used to assess the performance of the two NMF algorithms and gauge their robustness when supplied with noisy datasets. These metrics are: Relative Reconstruction Errors (RRE), Average Accuracy and Normalized Mutual Information (NMI). Through our experiments, we varied different parameters like rank, factor initialization technique, image scaling technique, noise types, noise percentage for both the NMF algorithms using both image datasets. After performing extensive analysis, we drew conclusions based on the performance of the algorithms with varied experimental setup in section 4.

2 Previous Work

A lot of variants have come up since Lee and Seung first introduced Frobenius NMF [4](discussed in section 3.3). Some of these algorithms are described as below:

2.1 Hypersurface cost based NMF

Hypersurface cost based NMF tries to minimize the value given by the hypersurface cost function for the original matrix and the estimated matrices [14]. Here the hypersurface cost function can be differentiated and can be solved by a method involving projected gradient [14]. The cost function for HCNMF is mentioned below [14] in equation 1:

$$h(x) = \sqrt{(1 + x^2) - 1} \quad (1)$$

Advantage: It is more robust compared to standard NMF (Frobenius norm)[15]

Disadvantage: It is difficult to optimize Hypersurface cost based NMF because it takes a lot of time to use the complex Armijo's rule-based line search [14].

2.2 L1 norm NMF

Lam proposed the L1 norm NMF algorithm which uses Laplace distribution to model the noise [16] (refer to equation 2).

$$\min_{D \geq 0, R \geq 0} \|X - DR\|_1, \text{ where } \|X\|_1 = \sum_{ij} |X_{ij}| \quad (2)$$

Advantage: It is resilient towards noise and outliers present in the data [17].

Disadvantage: Its optimization process proved to be computationally expensive as L1 norm-based NMF is not smooth and cannot be scaled to larger datasets as it is dependent on the number of attributes (dimensionality) of the data [17].

2.3 Manhattan NMF

Manhattan NMF tries to reduce the Manhattan distance between the original data and its approximations (refer to equation 3) such that the model performs well in spite of being a heavy tailed noise distribution [18].

$$\min_{D, R} \|X - DR\|_M \quad s.t. \quad D \geq 0, R \geq 0 \quad (3)$$

Advantage: It approximates the non-smooth(non-differentiable and discontinuous functions) loss function with a smooth one and uses Nesterov's method to minimize the approximated loss function.

This makes it more robust to outliers and noise as compared to L1-NMF [20].

Disadvantage: It cannot handle continuous occlusion implying that a robust subspace cannot be achieved for clustering [16].

2.4 L1 Regularized Robust NMF

In order to restore the non-corrupted data, Zhang et al. developed a method (refer to equation 4) which subtracted the noise (sparse error matrix) from the contaminated data. It is called L1-norm regularized Robust NMF [19]:

$$\min_{D, R, E} \|X - DR - E\|_F^2 + \lambda \|E\|_1 \quad s.t D \geq 0, R \geq 0. \quad (4)$$

Advantage: It is able to reconstruct the clean data from the corrupted data without knowing the location of the noise [19].

Disadvantage: It is non-robust to a large number of outliers due to a low breakdown point in L1-minimization model [17].

2.5 Robust Capped NMF

In order to eliminate the effect of outlier samples, Gao et al. proposed a method (refer to equation 5) which puts a constraint on their quantities in the formulation of the objective function [21]. It is called Robust capped NMF:

$$\sum_{D \geq 0, R \geq 0} \sum_{j=1}^n \min\|X_{.j} - DR_{.j}\|, \theta \quad (5)$$

where θ is the threshold value that selects the samples of outliers.

Advantage: It is more robust to outliers because it uses the outlier threshold value [21].

Disadvantage: There exists uncertainty in order to find the accurate value of the threshold for the outliers and hence, the utilized iterative algorithms have a slow convergence rate [17].

2.6 Truncated Cauchy based NMF

For truncated cauchy based NMF, the 3-sigma rule to trace outliers is given as [17] mentioned in equation 6 below :

$$\min_{D \geq 0, R \geq 0} F(D, R) = \sum_{i=1}^m \sum_{j=1}^n g\left(\frac{(X - DR)_{ij}}{\gamma}\right) \quad (6)$$

$$\text{where } g(x) = \begin{cases} \ln 1 + x, & 0 \leq x \leq \sigma \\ \ln 1 + \sigma, & x > \sigma \end{cases}$$

where σ is the scale parameter which is calculated using the 3-sigma rule and γ is the truncation parameter and is obtained by Nagy algorithm [17].

Advantage: Truncated Cauchy based NMF was developed to handle moderate and extreme outliers and become robust to noise by truncating the large errors [17].

Disadvantage: It is difficult for the NMF model to learn a subset of high-dimensional data which is resilient to noise and has fewer dimensions. Initially its performance is good, but later it starts decreasing when the noise percentage increases [15].

3 Methods

In this section, the pre-processing techniques are listed followed by the detailed formulation of NMF algorithms used in the assignment. Additionally, the definition of objective function and the optimization method are outlined. The types of  used in the experiment along with the analysis of the robustness of each algorithm has also been defined.

3.1 Exploratory Data Analysis

The first step towards implementing any machine learning algorithm is to understand the data properly. So both the image datasets were analysed thoroughly to understand the distribution of data, labels and other statistics.

To reduce the computational complexity, all the images were resized to a fixed dimension i.e. 30X37 pixels for ORL dataset and 168x192 pixels for YaleB dataset.

3.1.1 ORL dataset

On comprehending the distribution of labels for the images in ORL dataset, it was observed that each image had 10 labels and there were total 40 labels as shown in Fig 1. All the statistics related to this dataset is shown in Table 1.

Table 1: Exploratory data analysis of ORL dataset.

Number of labels	40
Number of images	400
Number of pixels in each image	1110
Dimension of ORL dataset	(1110, 400)
Dimension of ORL labels	(400,)
Minimum pixel value	9
Maximum pixel value	228

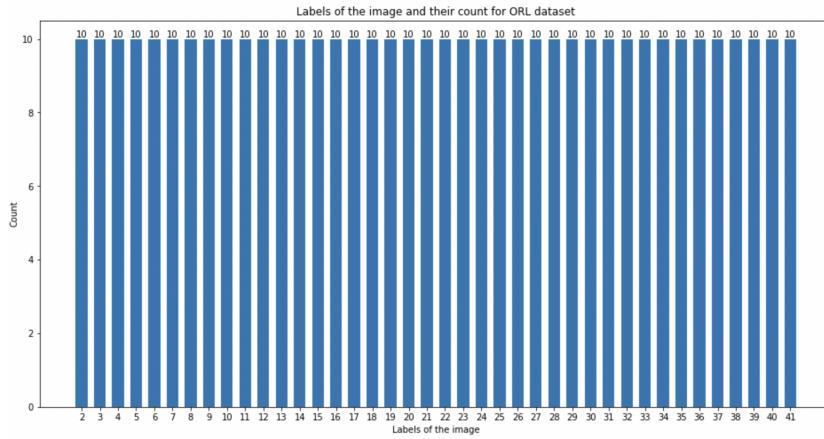


Figure 1: Labels of the image and their count for ORL dataset

3.1.2 YaleB dataset

On analysing the data for YaleB dataset, it was observed that the count of each label was not uniform. While most of the labels had count as 64, few had lesser count as shown in Fig 2). The general statistics for the dataset has been captured in Table 2.

3.2 Pre-processing

Three different image scaling techniques were used for pre-processing the data. A lot of tests were conducted to understand which scaling technique was more suitable for a particular type of NMF algorithm and its associated parameters. The details of the tests are mentioned in section 5. The methods used for image scaling were standardization, normalization, and centering(global and local). Fig 3 and Fig 4 shows sample images comparing original images with the pre-processed ones from ORL and YaleB dataset respectively.

Table 2: Exploratory data analysis of YaleB dataset.

Number of labels	38
Number of images	2414
Number of pixels in each image	2016
Dimension of YaleB dataset	(2016, 2414)
Dimension of YaleB labels	(2414,)
Minimum pixel value	0
Maximum pixel value	255

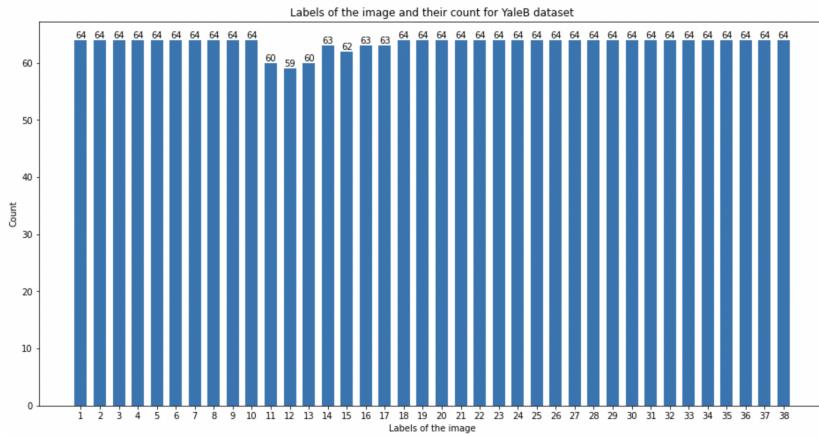


Figure 2: Labels of the image and their count for YaleB dataset

3.2.1 Standardization

The standardization method standardizes the image pixels for the given dataset for mean value 0 and standard deviation 1. The values were clipped to have only positive values.

3.2.2 Normalization

Normalization technique was used to scale the image pixels for the given dataset between the range of 0 to 1.

3.2.3 Centering

Centering centers the image pixels by subtracting the mean value from it. Both local and global centering techniques were used and then the result was standardized to ensure that the values are in positive range.

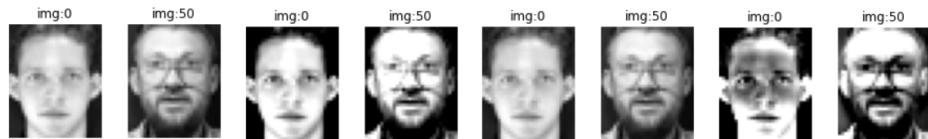


Figure 3: Sample images from ORL dataset showing original two images in the left, followed by images after standardization, followed by images after normalization and finally after centering and standardization

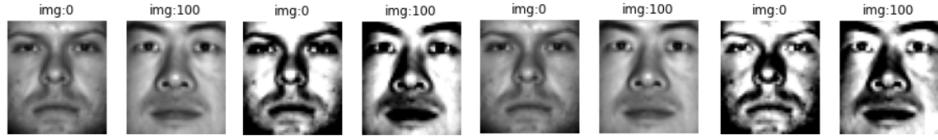


Figure 4: Sample images from YaleB dataset showing original two images in the left, followed by images after standardization, followed by images after normalization and finally after centering and standardization

3.3 Frobenius norm based NMF algorithm

Low rank approximations of images are known to be achieved by using non-matrix factorization. This quality of approximation is improved by using L2 based multiplicative update rules [22]. The repetition of iterations of the update rules assures the convergence to a locally optimal matrix factorization. One big disadvantage of the standard NMF is that the cost function uses a least squared function, which amplifies the value of the error in the presence of some outliers making it unstable with respect to noise and outliers [23]. NMF algorithms using Frobenius norm are known to be sensitive to non-Gaussian noise and outliers, which makes it difficult to reconstruct quality images.

3.3.1 Cost function

To decompose X into D and R , the cost function needs to be defined such that the quality of the approximation can be measured numerically. One way to do so is by squaring the Euclidean distance between X and the approximated matrices DR [24]. Formally, the cost function for L2 Frobenius norm based NMF is defined as mentioned in equation 7:

$$\min_{D \in \mathfrak{D}, R \in \mathfrak{R}} \|X - DR\|_F^2 \quad \text{Such that } D, R > 0 \quad (7)$$

3.3.2 Objective function

The cost function of L2 norm which guides the optimizer, is a NP-hard(nondeterministic polynomial time) problem as it is convex in D or R , but it is not convex in terms of both the variables together [22]. This implies that it can be generally approximated to find a local minimum rather than solving analytically to trace a global minimum. Although gradient descent is simple to implement, the convergence can be slow. Hence multiplicative update rules (refer to equations 8, 9, 10 and 11) are a viable option to deal with the trade-off between speed and computational complexity. The objective of this algorithm is to determine two matrices D and R , which can reduce the predefined error function.

Once D is fixed, we can update the cost function by using the partial derivative of R to update the function.

$$\frac{\partial \|X - DR\|_F^2}{\partial R} = -2D^T + 2D^T DR \quad (8)$$

Similarly, once R is fixed, we can update the cost function by using the partial derivative of D to update the cost function.

$$\frac{\partial \|X - DR\|_F^2}{\partial D} = -2R^T + 2R^T DR \quad (9)$$

These partial derivatives will help in optimizing the objective function through multiplicative update rules. The randomly sampled scaled matrices are initialized using uniform/Gaussian initialization techniques.

For k^{th} iteration, we fix D as D^k and solve for R^{K+1} ,

$$R_{i,j}^{k+1} = R_{i,j}^k \frac{(D^{k^T})_{i,j}}{D^{k^T} D^k R^{k^T}} \quad (10)$$

For k^{th} iteration, we fix R as R^{k+1} and solve for D^{k+1} ,

$$D_i^{k+1}, j = D_i^k, j \frac{X R^{k+1T}}{D^k R^{k+1} R^{k+1T}} \quad (11)$$

3.3.3 Advantages

Feature extraction is used to reduce the dimensions of the data space into a subspace with fewer dimensions. Methods like Principal Components Analysis (PCA), LDA(Linear Discriminant Analysis), Singular Value Decomposition (SVD), ICA (Independent Components Analysis) are successful in achieving this objective. However, PCA, SVD, LDA, ICA can provide negative components [25] which have no interpretation whereas the non-negative constraints of NMF decompose the matrix into pure non-negative matrices.

3.3.4 Disadvantages

There is no unique global minimum in NMF, like there is one in SVD [26]. The optimization problem is convex in either D or R but not in both [22]. Hence it only estimates the convergence to a local minimum. It is very sensitive to outliers and data contaminated with noise [26].

3.4 Robust Nonnegative Matrix Factorization using $L_{2,1}$ -norm

To overcome the challenges of Frobenius NMF, another variant of NMF called Robust Nonnegative Matrix Factorization using $L_{2,1}$ -norm was introduced which shows more resilience towards data corrupted by noise and outliers. Research has shown that $L_{2,1}$ NMF is more effective for convergence on real-image datasets as it has lower convergence error. Also, $L_{2,1}$ outperforms K-means and standard NMF during clustering analysis.

3.4.1 Cost function

The computational algorithm is simple and produces better clustering results as compared to standard NMF [27]. The cost function (refer to equation 12) contains an error term which is not squared and hence the outliers don't dominate the objective function. The cost function of Robust Nonnegative Matrix Factorization using $L_{2,1}$ -norm can be defined as:

$$\|X - DR\|_{2,1} = \sum_{i=1}^n \sqrt{\sum_{j=1}^p (X - DR)_{ji}^2} = \sum_{i=1}^n \|x_i - Dg_i\| \quad (12)$$

3.4.2 Optimization function

Just like standard NMF, the loss function of $L_{2,1}$ norm adapts the multiplicative update rules. A diagonal matrix Z_{ii} is calculated to perform regularization for the outliers in the features. The elements of the diagonal matrix Z (refer to equation 15) are weights that are used to suppress the effect of outliers. When we fix D and update R in the rule (refer to equation 13), the Karush-Kohn-Tucker condition holds true, and the objective function decreases monotonically. The same condition holds true when we fix R and update D (refer to equation 14).

$$D_{jk} \leftarrow D_{jk} \frac{(XZR^T)_{jk}}{(DRZR^T)_{jk}} \quad (13)$$

$$R_k i \leftarrow R_k i \frac{(D^T XZ)_{ki}}{(D^T DRZ)_{ki}} \quad (14)$$

where Z is a diagonal matrix. Below is the representation of the diagonal elements of matrix Z:

$$Z_{ii} = 1 / \sqrt{\sum_{j=1}^p (X - DR)_{ji}^2} = 1 / \|x_i - Dg_i\| \quad (15)$$

3.4.3 Advantages

The L₁ norm is more robust than Frobenius NMF to handle the outliers and noise and it also consists of better and sophisticated updating rules which can extend its application in several domains [27].



3.4.4 Disadvantages

One major drawback of the L_{2,1} norm is that its loss function is not smooth. Hence it consumes more time in achieving factorization [28].

3.5 Noise

One of the important criteria to test the robustness of NMF algorithm is to run it against different types of noise with varying noise percentage. Since the algorithms chosen for this assignment were Frobenius and L_{2,1} norm NMF, the noise types chosen for the experiments were Gaussian, Laplace and Impulse to cover the range of noises where these algorithms work best, where they are sensitive and for additional testing.

3.5.1 Gaussian Noise

Gaussian noise in general causes the distortion of gray values in the images[29]. Designed and characterized by its probability density function (PDF), Gaussian noise model (refer to equation 16) forms a bell curve (normalized histogram) with respect to gray value. The mean and standard deviation of the image dataset was used to produce random Gaussian noise in our experiments [30]. The formula is mentioned below:

$$P(g) = \sqrt{\frac{1}{2\pi\sigma^2}} e^{-\frac{(g-\mu)^2}{2\sigma^2}} \quad (16)$$

where g= gray image, σ = standard deviation and μ = mean. The Gaussian distribution function projects a peak at its mean and its distribution is more widespread as the standard deviation increases.

3.5.2 Laplace Noise

Laplace noise is obtained from Laplacian distribution which is thinner at the peak and flatter at the tails [31]. Given a distribution of exponential variables generated randomly and are identical in nature, Laplacian noise represents the difference between those variables [32]. The probability density function of Laplace noise is as mentioned below in equation 17:

$$f(x; \mu; \lambda) = \frac{1}{2\lambda} \exp(-\frac{|x - \mu|}{\lambda}) \quad (17)$$

where lambda = exponential decay which must be non-negative and μ = mean. Laplacian distribution is deemed to be better than Gaussian distribution in various sectors such as economics and health sciences. A lot of values of lambda were tried to generate proper noise for different images like original and scaled ones. Based on the test runs, different values of lambda were selected for each scaling type.

3.5.3 Impulse Noise

Impulse noise is also referred as salt and pepper noise. It changes the quality of the original image data. In this case salt noise sets the pixel value to 255 and pepper noise sets the value to zero [29]. Based on the noise percentage, the pixels to be changed are randomly selected. Half of those pixels are updated with salt and the other half is updated by pepper noise.

Fig 5 shows the sample image with no noise followed by 50% noise Gaussian, Laplace and Impulse from both datasets ORL and YaleB respectively.

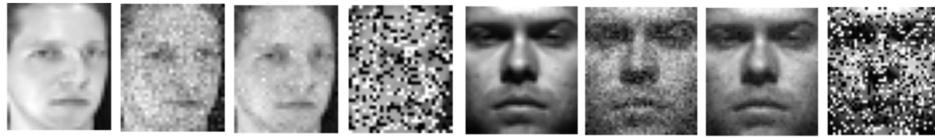


Figure 5: Sample image from ORL dataset showing original image in the left, followed by image with 50% Gaussian Noise, followed by image with 50% Laplace Noise and finally image with 50% Impulse Noise. The same information is shown for YaleB dataset from 5th image from the left.

4 Experiments and Discussions

Two datasets ORL and YaleB were used for conducting the experiments. The datasets have been described in detail in section 3.1. All the experiments were conducted on these two datasets using Frobenius norm NMF and $L_{2,1}$ norm NMF. Each test was conducted for 5 times and the average and standard deviation was calculated for the result. Repeating these tests multiple times ensures the robustness of the performance of the NMF algorithm in general. The details of the experiments are described in section 4.1 and 4.2.

4.1 Experiments, comparisons and evaluation

There are a few factors which are key to getting optimal results from NMF algorithms apart from the algorithm itself. They are:

- Choice of rank of the matrix X to factorize it to D and R.
- Initialization technique of the factors (D, R), of the matrix
- Image scaling technique
- Type of noise and noise percentage

One of the obvious choices to find the right set of hyper-parameters for an algorithm was to use grid search. As grid search is computationally very expensive, random search [34] was used in this assignment. Exhaustive tests were conducted to understand the impact of each of the hyper-parameter. Ablation study was also done to understand the factors impacting the performance of the algorithm. The details of the key tests and its results have been listed in section 4.2.

Apart from these factors, stopping criteria of the algorithm is equally important [35]. In this assignment, a combination of stopping criteria was used to get the optimal result in less computational time. A lot of tests were conducted using combinations of fixed number of iterations and error threshold based stopping criteria. Based on the result, a generous but fixed number of iterations was chosen.

Relative Reconstruction Errors(RRE), Average Accuracy and Normalized Mutual Information(NMI) were used to evaluate the result of each experiment. This covers different aspects of the robustness of the algorithm.

4.2 Extensive analysis and discussion of results

4.2.1 Experiments for Rank of the factors

The most important step was to determine the rank of the factors (D,R) such that the image reconstruction is acceptable and at the same time we are able to get the benefit of image compression. The general guidance for selecting k is that it should be very less than n. Given X is m*n matrix and D and R are m*k and k*n matrix respectively, we tried some calculations to understand the compression ratio for different values of k.

$$\begin{aligned} X_{\text{ORL}} (m \times n) &= 1110 * 400 \\ 90\% \text{ sample of } X_{\text{ORL}}(m \times n) &= 1110 * 360 \end{aligned}$$

$$X_{\text{YaleB}} (m \times n) = 2016 * 2414$$

90% sample of X_YaleB ($m \times n$) = 2016 * 2173

Table 3 and Table 4 lists some of the compression ratios for various values of k for ORL and YaleB dataset respectively.

Table 3: Compression ratio for ORL dataset

k	Compression ratio ($k(m + n)/(m \times n)$)
2	0.73%
50	18.39%
100	36%
150	50%

Table 4: Compression ratio for YaleB dataset

k	Compression ratio ($k(m + n)/(m \times n)$)
2	0.19%
100	9.5%
300	28%
400	38%

A lot of experiments were conducted to understand the impact of the value of k on the reconstruction of image. The parameters listed above table 5 were kept fixed and k was changed to understand the impact of it on the overall result for both Frobenius and $L_{2,1}$ norm based NMF on both datasets ORL and YaleB.

Fixed Parameters:

Noise Type: Gaussian

Noise Percentage: 50%

Image Scaling: No Scaling

Factor initialization Type: Uniform

Table 5: Experiments for Rank of the factors

Parameters	Ranks	Observation
NMF Algo: Frobenius/ $L_{2,1}$ Dataset: ORL	[2, 5, 25, 50, 75, 100]	RRE was lowest in the case of rank 100. Average Accuracy and NMI converged to similar numbers for ranks over 5 with very marginal high peaks for rank 25. The standard deviation was very less and more stable for rank 100 than 25. So 100 was chosen as the rank for further experiments.
NMF Algo: Frobenius/ $L_{2,1}$ Dataset: YaleB	[2, 100, 300, 400]	RRE was lowest in the case of rank 400 especially with $L_{2,1}$ norm. Average Accuracy and NMI converged to similar numbers for ranks 300 and 400 with very marginal high peaks for rank 100. The standard deviation was almost similar for 300 and 400 with rank 300 having slightly better SD for accuracy and nmi. For almost the same result the compression ratio was 10% more in rank 400 than 300 so rank 300 was chosen. For $L_{2,1}$ norm, 300 and 400 ranks were tried and 300 was selected for similar reason like that of Frobenius norm.

The above results were also validated by manually checking the reconstructed images against the original images and the noisy images. For the ORL dataset, the compression ratio was 36% with $k=100$ and the reconstructed image was also quite clear, so 100 was chosen as the rank for further experiments. For YaleB dataset, the compression ratio was 28% with $k=300$ and the reconstructed image was also quite clear, so 300 was chosen as the rank for further experiments. Fig 6 shows

the image comparison for rank 100 for ORL and rank 300 for YaleB dataset. Refer to section 3 Experiment Setup in the code for detailed results. The comparative graph for average value of RRE, Accuracy and NMI and its standard deviation for various ranks in case of ORL dataset with Frobenius NMF is shown in Fig 7.

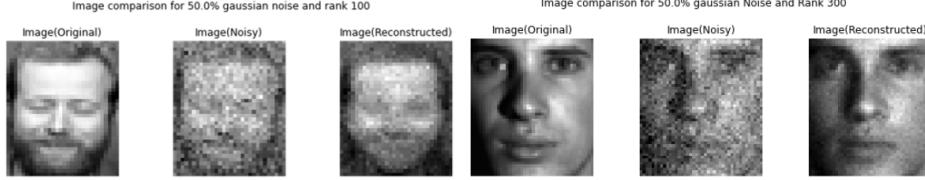


Figure 6: Image comparison for rank 100 for ORL and rank 300 for YaleB dataset.

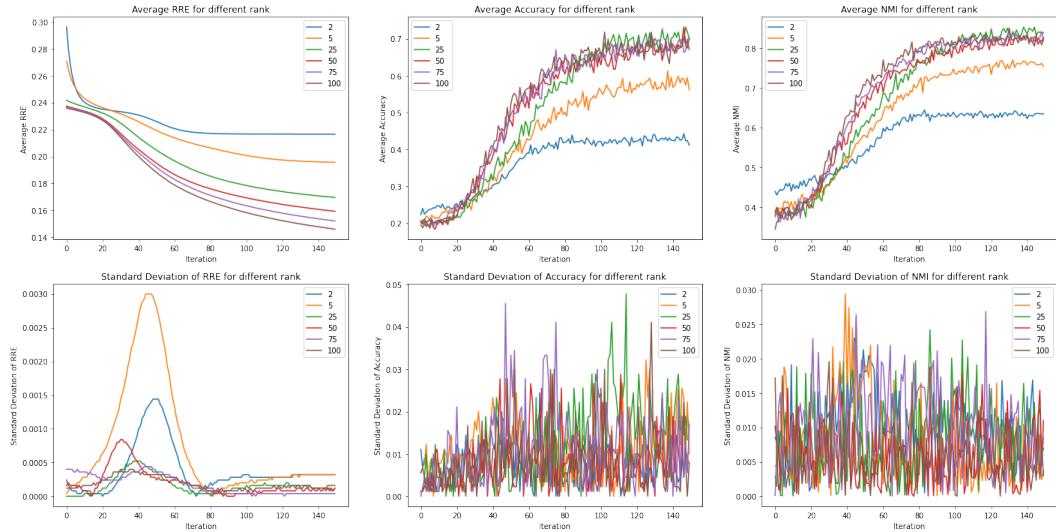


Figure 7: Average values of RRE, Accuracy and NMI and its standard deviation for various ranks for ORL dataset with Frobenius NMF.

4.2.2 Experiments for Initialization of the factors

Factor initialization is a very important step in NMF algorithm as it helps in reaching the optimal result in less computational cost [36]. There are different techniques using which factors can be initialized. In this assignment, Random based techniques like Uniform and Gaussian initialization methods were used. These methods are low in computational cost as compared to other methods. Table 6 describes the experimental setup and observations related to factor initialization type.

Fixed Parameters:

Noise Type: Gaussian

Noise Percentage: 50%

Image Scaling: No Scaling

Based on the observations and reconstructed images, the Uniform method was selected for further analysis. Fig 8 shows the comparison of sample images from YaleB dataset for Uniform and Gaussian factor initialization technique. The reconstruction of the images was tested with 50% Gaussian noise.

Table 6: Experiments for Initialization of the factors

Parameters	Factor Initialization Type	Observations
NMF Algo: Frobenius Dataset: ORL/YaleB Rank: 100	[uniform, gaussian]	The Uniform method was better than Gaussian for both NMF algorithms. In the case of Frobenius NMF, the Uniform method did have a very minor spike in case of standard deviation of the average accuracy.
NMF Algo: $L_{2,1}$ Dataset: ORL/YaleB Rank: 300	[uniform, gaussian]	

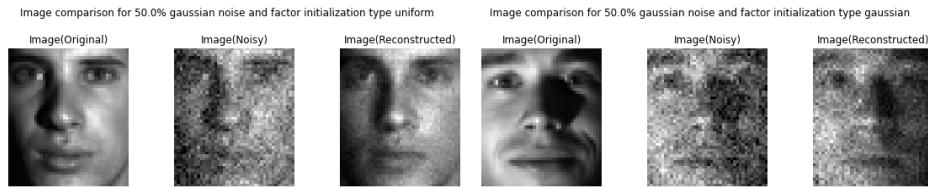


Figure 8: Image comparison for Uniform vs Gaussian initialization technique on sample images from YaleB dataset using Frobenius norm

4.2.3 Experiments for Image Scaling Techniques

Image scaling plays a vital role in the calculations of the NMF algorithm. In this assignment 3 types of image scaling techniques were used and they were also compared with the original image without any scaling. Table 7 lists the tests conducted for image scaling.

Fixed Parameters:

Noise Type: Gaussian

Noise Percentage: 50%

Factor Initialization: Uniform

Table 7: Experiments for Image Scaling Techniques

Parameters	Image Scaling Type	Observations
NMF Algo: Frobenius Dataset: ORL/YaleB Rank: 100	[no scaling, standardization, normalization, centering]	In the case of Frobenius Algorithm, original data and normalized data performed almost equally well on both datasets but original data had little more deviation than normalized data. Centered data was little better in terms of accuracy but took a dip in case of RRE and NMI. $L_{2,1}$ was more robust with image centering technique for both datasets.
NMF Algo: $L_{2,1}$ Dataset: ORL/YaleB Rank: 300	[no scaling, standardization, normalization, centering]	

4.2.4 Experiments for Noise Types

Various noise techniques like Gaussian, Laplace and Impulse(Salt and Pepper), were used to understand the behaviour of the type of noise on the learning process of matrix factors, D and R. Some experiments were conducted to understand how the noise will be generated based on the hyper-parameters for these noise types and image scaling techniques. The hyper-parameters were tuned and used accordingly. For example, in case of Laplace noise, the value of lambda determines how the noise will be generated when the image has no scaling or has been standardized, normalized or centered. Table 8 lists the experiment setup related to different noise types. Fig 9 shows 25 images from basis vector (D) for ORL dataset with $L_{2,1}$ norm and Gaussian noise(50% Gaussian noise).

Fixed Parameters:

Noise Percentage: 50%

Factor Initialization: Uniform

Table 8: Experiments for Noise Technique

Parameters	Noise Technique	Observations
NMF Algo: Frobenius Dataset: ORL/YaleB Image Scaling: Normalization Rank: 100	[gaussian, laplace, impulse]	In case of Frobenius norm, Impulse noise had very less RRE value but it had significant low performance for average accuracy and NMI with respect to Gaussian and Laplace noise type. Impulse noise had less deviation as compared to the other two types. Laplace and Gaussian had similar performance but Laplace was slightly better. For $L_{2,1}$ norm, Gaussian was a bit less robust than Laplace in terms of RRE but was much better in accuracy and NMI. It did show some small deviation as compared to Laplace.
NMF Algo: $L_{2,1}$ Dataset: ORL/YaleB Image Scaling: Centering Rank: 300	[gaussian, laplace, impulse]	



Figure 9: 25 images from basis vector (D) for ORL dataset with $L_{2,1}$ norm and 50% Gaussian noise

4.2.5 Experiments for Noise Percentage

Different percentage of noise was used to test the robustness of the algorithm. Table 9 below lists the details of the test runs.

Fixed Parameters:

Factor Initialization: Uniform

Table 9: Experiments for Noise Percentage

Parameters	Noise Percentage	Observations
NMF Algo: Frobenius Dataset: ORL/YaleB Image Scaling: Normalization Rank: 100 Noise Technique: Laplace	[0, 20, 50, 70, 90]	The algorithms performed well on almost all percentage of the noise
NMF Algo: L21 Dataset: ORL/YaleB Image Scaling: Centering Rank: 300 Noise Technique: Gaussian	[0, 20, 50, 70, 90]	

Fig 10 and Fig 11 shows the graph for average and standard deviation of RRE, Accuracy, NMI for different noise percentage for ORL dataset using Frobenius norm and L21 NMF respectively.

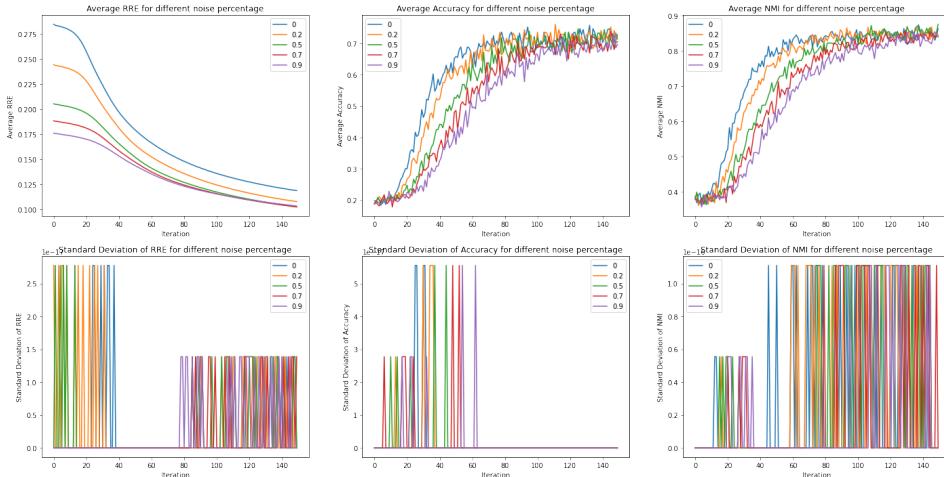


Figure 10: Graph for average and standard deviation of RRE, Accuracy, NMI for ORL dataset using Frobenius norm NMF for different noise percentage

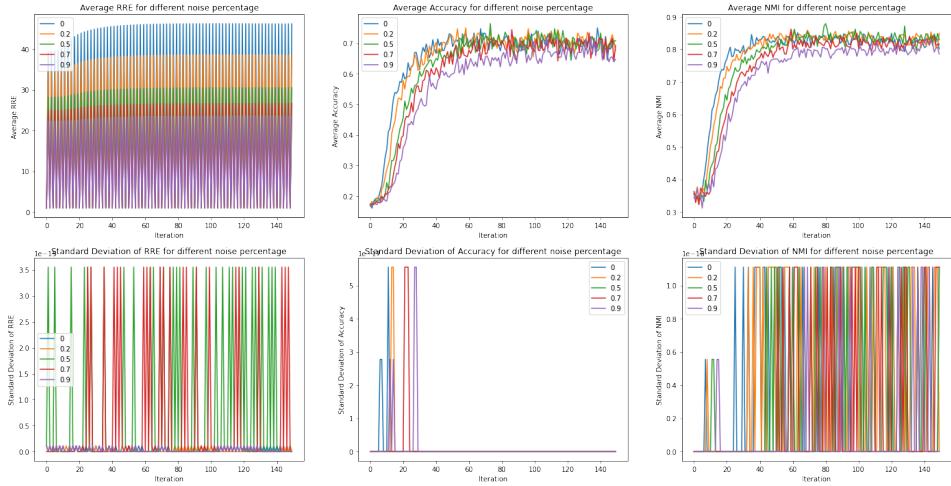


Figure 11: Graph for average and standard deviation of RRE, Accuracy, NMI for ORL dataset using L21 norm NMF for different noise percentage

4.2.6 Experiments for different NMF algorithms

One more set of test was done to compare the two algorithms directly on a given set of parameters on both the datasets. Table 10 below lists the details of the test runs.

Fixed Parameters:

Factor Initialization: Uniform
Noise Technique: Gaussian
Noise percentage: 70%

Table 10: Experiments for NMF types

Parameters	NMF Type	Observations
Dataset: ORL Image Scaling: Normalization Rank: 100	[Frobenius, L21]	The reconstructed images were almost similar using both the algorithms for a given set of parameters. There was a big fluctuation in the value of average RRE for L21 norm for both datasets. The average accuracy and NMI seemed to be similar in behaviour for both datasets for both algorithms. Though there was a difference in the max range for RRE, Accuracy and NMI for both the datasets.
Dataset: YaleB Image Scaling: Centering Rank: 300	[Frobenius, L21]	

Fig 12 and Fig 13 shows the comparison of original image, noisy image and the reconstructed image using both NMF algorithms for ORL and YaleB dataset respectively. Fig 14 and Fig 15 shows the graph for average and standard deviation of RRE, Accuracy, NMI for both the NMF algorithms for ORL and YaleB dataset respectively.

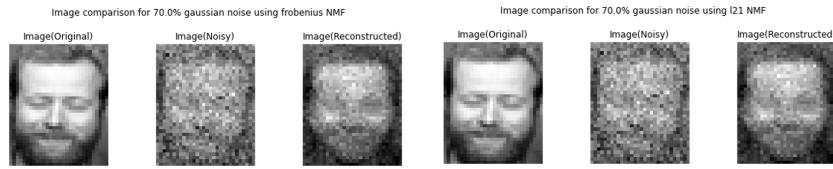


Figure 12: Image reconstruction comparison for Frobenius and L21 norm NMF on ORL dataset

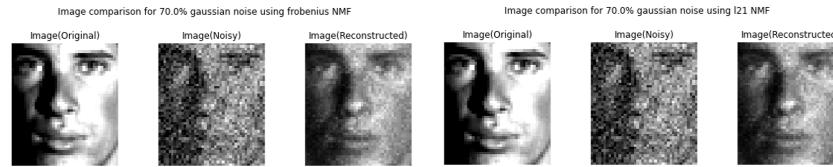


Figure 13: Image reconstruction comparison for Frobenius and L21 norm NMF on YaleB dataset

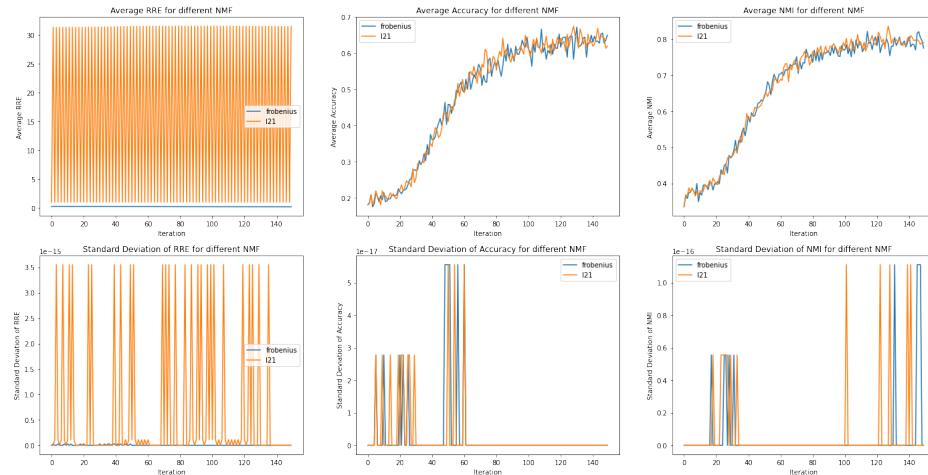


Figure 14: Avg and SD of RRE, Accuracy, NMI for ORL dataset using Frobenius and L21 NMF

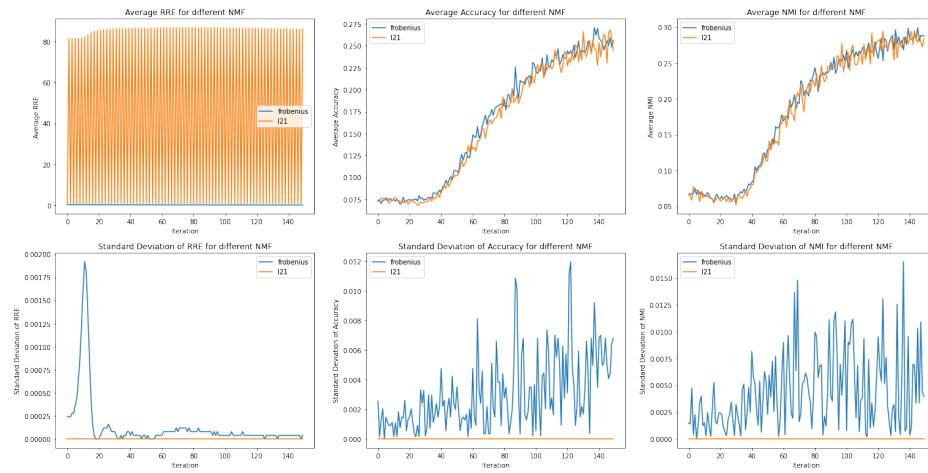


Figure 15: Avg and SD of RRE, Accuracy, NMI for YaleB dataset using Frobenius and L21 NMF

4.3 Relevant personal reflection

The Table 11 shows a comparative analysis of the min and max values for mean of RRE, ACC and NMI for the experiment setup explained in Table 10. It indicates that even though the reconstructed images are quite similar in nature from both algorithms on both datasets, the overall RRE, ACC and NMI are having a good variation for different datasets using different NMF algorithms. It was also observed that the overall accuracy and NMI was low and RRE was relatively high for YaleB dataset as compared to ORL dataset.

Table 11: Analysis of result for ORL and YaleB dataset for both NMF algorithms

Dataset	NMF Type	Min Avg RRE	Max Avg Acc	Max Avg NMI
ORL	Frobenius	0.1634	0.6722	0.8205
ORL	L21	0.9704	0.675	0.8351
YaleB	Frobenius	0.2784	0.2705	0.2995
YaleB	L21	0.9898	0.2678	0.296

5 Conclusions and Future work

5.1 Conclusion

In order to analyse the robustness of NMF algorithms on ORL and YaleB datasets, different hyper-parameters were tried. The following inferences were made after extensive research and experimentation:

Rank: As we increased the value of rank (k), the algorithm delivered clearer reconstructed images from the noisy ones. Rank (k=100) and (k=300) were found to be the best choices for the ranks for ORL and YaleB datasets respectively since they had the benefit of both the worlds, low storage and high quality of reconstructed images.

Factor Initialization: Between Gaussian and Uniform method for factor initialization, Uniform method delivered best result for both the datasets for both Frobenius and L21 NMF algorithm.

Image Scaling: Frobenius algorithm worked better with normalized images whereas L21 had better performance with images scaled using centering technique.

Noise Type: Laplace and Gaussian noise types had almost similar performance in case of Frobenius NMF whereas for L21, Gaussian performed better.

Noise percentage: Both Frobenius and L21 algorithms performed well with different values of noise percentage.

5.2 Future work

These experiments lead to a lot of research opportunities in order to analyse the robustness of two NMF algorithms with noisy images. The following can be explored in future work:

- The reason for variation in the value of RRE for L21 norm should be explored further as it was observed that the range of variation was quite high as compared to Frobenius norm.
- More random combination of the hyper-parameters should be tested to understand the impact of these parameters better.

6 Appendix

6.1 System Configuration

The hardware and software requirement is based on the laptop where the code was developed and tested.

- Hardware requirements- Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz 1.50 GHz
- Software requirements- Windows 10, Python 3

6.2 Instructions to execute the code

The code was developed and tested on a personal laptop and not on google colab. Follow the steps mentioned below to execute the code:

- The default folder path for the datasets are set as explained in the instructions of the Assignment. If it needs to be changed, update the paths in section 1.0 Data Folder in Assignment1.ipynb.
- Execute all the cells till section 1.1 to load both the datasets.
- Execute all cells in section 1.3 and its subsection for Initial data analysis
- Execute all cells in section 1.4 and its subsection for various pre-processing techniques
- Execute all cells in section 2.1, 2.2, 2.3 and 2.4 and 2.5 for Random sampling of data, Noise addition, Factor initialization, Evaluation metrics and NMF algorithms.
- Execute 3 cells in section 3 (before 3.1) for experiment setup and result plotting code.
- All cells in 3.1 section and subsection are possible setups of experiments related to ORL dataset. The details of the experiments are mentioned in a tabular format. The hyperparameters are well documented and all possible values are listed. The values can be changed to test any possible combination of the hyper parameters for any of the experiments. The same case is with all cells in 3.2 section and its subsection for YaleB dataset.
- Section 5 is for additional Tests Plots for more adhoc testing.

6.3 Contribution of the team members

Both the team members contributed equally to the assignment.

References

- [1]S. Alasadi and W. Bhaya, "Review of Data Preprocessing Techniques in Data Mining", Medwelljournals.com, 2017. [Online]. Available: <https://medwelljournals.com/abstract/?doi=jeasci.2017.4102.4107>. [Accessed: 12- Oct- 2021].
- [2]L. Sunithaa, M. Bal Rajua and B. Sunil Srinivasa, "A Comparative Study between Noisy Data and Outlier Data in Data Mining", International Journal of Current Engineering and Technology, vol. 3, no. 2, 2013. Available: <https://inpressco.com/wp-content/uploads/2013/06/Paper64575-577.pdf>. [Accessed 12 October 2021].
- [3]M. Rezaei, R. Boostani and M. Rezaei, "An Efficient Initialization Method for Nonnegative Matrix Factorization", Journal of Applied Sciences, vol. 11, no. 2, pp. 354-359, 2011. Available: 10.3923/jas.2011.354.359.
- [4]D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization", Nature, vol. 401, no. 6755, pp. 788-791, 1999. Available: 10.1038/44565.
- [5]Z. Chen and A. Cichocki, "Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints", Laboratory for Advanced Brain Signal Processing, RIKEN, Tech. Rep, 2005. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.63.7094>. [Accessed 10 October 2021].
- [6]L. Zhao, G. Zhuang and X. Xu, "Facial expression recognition based on PCA and NMF", 2008 7th World Congress on Intelligent Control and Automation, pp. 6826-6829, 2008. Available: 10.1109/wcica.2008.4593968 [Accessed 9 October 2021].
- [7]M. Berry, M. Browne, A. Langville, V. Pauca and R. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization", Computational Statistics & Data Analysis, vol. 52, no. 1, pp. 155-173, 2007. Available: 10.1016/j.csda.2006.11.006 [Accessed 7 October 2021].
- [8]A. Bougacha et al., "Rank-Two NMF Clustering for Glioblastoma Characterization", Journal of Healthcare Engineering, vol. 2018, pp. 1-7, 2018. Available: 10.1155/2018/1048164 [Accessed 12 October 2021].
- [9]G. Naik, "Non-negative Matrix Factorization Techniques", Signals and Communication Technology, 2016. Available: 10.1007/978-3-662-48331-2 [Accessed 12 October 2021].

- [10]S. Li, X. Wen Hou, H. Jiang Zhang and Q. Sheng Cheng, "Learning spatially localized, parts-based representation", Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001. Available: 10.1109/cvpr.2001.990477 [Accessed 12 October 2021].
- [11]D. Guillamet and J. Vitrià, "Non-negative Matrix Factorization for Face Recognition", Lecture Notes in Computer Science, pp. 336-344, 2002. Available: 10.1007/3-540-36079-4_29 [Accessed 12 October 2021].
- [12]Z. Guo and Y. Zhang, "A Sparse Corruption Non-Negative Matrix Factorization method and application in face image processing & recognition", Measurement, vol. 136, pp. 429-437, 2019. Available: 10.1016/j.measurement.2018.12.087 [Accessed 12 October 2021].
- [13]M. Berry, M. Browne, A. Langville, V. Pauca and R. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization", Computational Statistics & Data Analysis, vol. 52, no. 1, pp. 155-173, 2007. Available: 10.1016/j.csda.2006.11.006 [Accessed 12 October 2021].
- [14]A. Hamza and D. Brady, "Reconstruction of reflectance spectra using robust nonnegative matrix factorization", IEEE Transactions on Signal Processing, vol. 54, no. 9, pp. 3637-3642, 2006. Available: 10.1109/tsp.2006.879282.
- [15]X. Dai, N. Zhang, K. Zhang and J. Xiong, "Weighted Nonnegative Matrix Factorization for Image Inpainting and Clustering", International Journal of Computational Intelligence Systems, vol. 13, no. 1, p. 734, 2020. Available: 10.2991/ijcis.d.200527.003 [Accessed 12 October 2021].
- [16]E. Lam, "Non-negative matrix factorization for images with Laplacian noise", APCCAS 2008 - 2008 IEEE Asia Pacific Conference on Circuits and Systems, pp. 798-801, 2008. Available: 10.1109/apccas.2008.4746143 [Accessed 12 October 2021].
- [17]N. Guan, T. Liu, Y. Zhang, D. Tao and L. Davis, "Truncated Cauchy Non-Negative Matrix Factorization", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 1, pp. 246-259, 2019. Available: 10.1109/tpami.2017.2777841.
- [18]N. Guan, D. Tao, Z. Luo and J. Shawe-Taylor, "MahNMF: Manhattan Non-negative Matrix Factorization", arXiv.org, 2021. [Online]. Available: <https://arxiv.org/abs/1207.3438>. [Accessed: 12- Oct- 2021].
- [19]L. Zhang, Z. Chen, M. Zheng and X. He, "Robust non-negative matrix factorization", Frontiers of Electrical and Electronic Engineering in China, vol. 6, no. 2, pp. 192-200, 2011. Available: 10.1007/s11460-011-0128-0.
- [20]Y. Nesterov, "Smooth minimization of non-smooth functions", Mathematical Programming, vol. 103, no. 1, pp. 127-152, 2004. Available: 10.1007/s10107-004-0552-5.
- [21]H. Gao, F. Nie, W. Cai and H. Huang, "Robust Capped Norm Nonnegative Matrix Factorization", Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015. Available: 10.1145/2806416.2806568 [Accessed 13 October 2021].
- [22]D. Lee and H. Seung, "Algorithms for non-negative matrix factorization", Proc NIPS, pp. 556-562, 2000.
- [23]W. Liu, N. Zheng and Q. You, "Nonnegative matrix factorization and its applications in pattern recognition", Chinese Science Bulletin, vol. 51, no. 1, pp. 7-18, 2006. Available: 10.1007/s11434-005-1109-6 [Accessed 13 October 2021].
- [24]P. Paatero and U. Tapper, "Least squares formulation of robust non-negative factor analysis", Chemometrics and Intelligent Laboratory Systems, vol. 37, no. 1, pp. 23-35, 1997. Available: 10.1016/s0169-7439(96)00044-5.
- [25]A. Limem, G. Delmaire, M. Puigt, G. Roussel and D. Courcot, "Non-negative Matrix Factorization under equality constraints—a study of industrial source identification", Applied Numerical Mathematics, vol. 85, pp. 1-15, 2014. Available: 10.1016/j.apnum.2014.05.009 [Accessed 11 October 2021].
- [26]R. Albright, J. Cox, D. Duling, A. Langville and C. Meyer, "Algorithms, initializations, and convergence for the nonnegative matrix factorization", Proc. 12th ACM SIGKDD Int. Conf. Knowl. Disc. & Data Mining, 2014. [Accessed 10 October 2021].
- [27]D. Kong, C. Ding and H. Huang, "Robust nonnegative matrix factorization using L21-norm", Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11, pp. 673–682, 2011. Available: 10.1145/2063576.2063676 [Accessed 13 October 2021].
- [28]X. Dai, N. Zhang, K. Zhang and J. Xiong, "Weighted Nonnegative Matrix Factorization for Image Inpainting and Clustering", International Journal of Computational Intelligence Systems, vol. 13, no. 1, p. 734, 2020. Available: 10.2991/ijcis.d.200527.003 [Accessed 13 October 2021].
- [29]A. Boyat and B. Joshi, "A Review Paper: Noise Models in Digital Image Processing", Signal & Image Processing : An International Journal (SIPIJ), vol. 6, no. 2, 2015. [Accessed 10 October 2021].

- [30]P. R. Peebles Jr., "Central Limit Theorem" in "Probability, Random Variables and Random Signal Principles", 4th ed., 2001, pp. 51, 51, 125. [Accessed 9 October 2021].
- [31]M. Abramowitz and I. Stegun, Handbook of mathematical functions with formulas, graphs, and mathematical tables. New York: Dover, 1971.
- [32]S. Kotz, T. Kozubowski and K. Podgórski, "The Laplace Distribution and Generalizations", 2001. Available: 10.1007/978-1-4612-0173-1 [Accessed 13 October 2021].
- [33]M. Koli and B. S, "Literature Survey on Impulse Noise Reduction", *Signal & Image Processing : An International Journal*, vol. 4, no. 5, pp. 75-95, 2013. Available: 10.5121/sipij.2013.4506.
- [34]A. Ng, "Hyperparameter Tuning in Practice (C2W3L03)", Youtube.com, 2017. [Online]. Available: <https://www.youtube.com/watch?v=wKkcBPp3F1Y>. [Accessed: 08- Oct- 2021].
- [35]S. ESSID and A. OZEROV, "A TUTORIAL ON NONNEGATIVE MATRIX FACTORISATION WITH APPLICATIONS TO AUDIOVISUAL CONTENT ANALYSIS", Perso.telecom-paristech.fr, 2014. [Online]. Available: https://perso.telecom-paristech.fr/essid/teach/NMF_tutorial_ICME-2014.pdf. [Accessed: 07- Oct- 2021].
- [36]F. Esposito, "A Review on Initialization Methods for Nonnegative Matrix Factorization: Towards Omics Data Experiments", *Mathematics*, vol. 9, no. 9, p. 1006, 2021. Available: 10.3390/math9091006 [Accessed 10 October 2021].