Robust 2D Principal Component Analysis: A Structured Sparsity Regularized Approach

Implementation Track, 4 Team Members

Abstract

In this paper, we attempted to re-implement and evaluate the performance of a robust principal component analysis (PCA) model for two-dimensional images regularized using structured sparse priors. The work is based on "Robust 2D Principal Component Analysis: A Structured Sparsity Regularized Approach" [16]. Compared to the traditional one-dimensional formulation of PCA, this model is able to better capture the spatial structure of the original images. In addition, the structured sparsity regularization allows the model to be more robust to outliers and noise in the data. As the objective function is non-convex and non-smooth, a two-stage alternating optimization method was used to solve the problem. We evaluated the performance of the algorithm through an image reconstruction study; however, We were unable to match the results presented in the original paper. We believe that this was due to an implementation difference when solving a part of the optimization problem.

1 Introduction

Principal component analysis (PCA) is a ubiquitous dimensionality reduction technique in machine learning and is commonly used for computer vision along with image and video processing tasks. A fundamental limitation however, lies in the need for the vectorization of the data into a single long vector, thus eliminating structural priors in the original data. A natural solution to this is to perform PCA directly on the data in its original format. This goal has led to the development of two-dimensional principal component analysis (2D-PCA) and other higher-order tensor decomposition methods that allow the algorithm to exploit the preexisting structural priors available in the data for a more efficient and accurate low-dimensional representation.

In this project, we re-implemented the structured sparsity regularization 2D-PCA algorithm proposed by Sun et al. [16] and compared its effectiveness against ordinary PCA (1D-PCA) on the Yale Face Database. Given the importance of PCA in modern-day machine learning applications, we think that this re-implementation is worth pursuing in order to understand the limitations of PCA that we are most familiar with. In addition, our comparisons of the two algorithms might provide insight into the importance of carefully choosing and/or constructing regularization methods for PCA and other algorithms.

2 Related Work

As stated by Sun et al. [16], sparse representation, low-rank matrix factorization, and robust PCA have been increasingly popular in face recognition [21, 17], background subtraction [6, 2, 22], image compression [1], visual object tracking [13], and image classification [12, 3]. These applications used various techniques to improve the robustness of their models such as l_0- and l_1-norm regularization, rank minimization, and trace-norm surrogate amongst others. A key assumption in all of these methods is that noise and errors in the data are sparsely located at random – an assumption that is often not true. For example, in applications with partial occlusions of faces or moving objects in the

foreground, outliers in the data are sparsely clustered spatially and/or temporally, and thus cannot be handled by the previously mentioned regularization methods. Some robust PCA approaches like proximal gradient descend [20] and alternating direction method of augmented Lagrange multiplier [10] are developed to improve the tolerance to errors with main ideas of rank minimization and convex substitutions.

Several algorithms have been proposed to incorporate the structural priors of the data, like group variables and feature selection [23], group lasso for logistic regression [14], structured sparsity in PCA [7]; however, they require the signals to be one-dimensional, effectively eliminating spatial correlations that may have existed in the data. In addition, the vectorization of images, and, even more so, with video frames, substantially increases the memory required and quickly becomes infeasible to process on smaller systems. Given the importance of PCA for image and video processing applications, the main contribution of Sun et al. [16] is the design of a two-dimensional PCA model capable of retaining the spatial correlations in the original data by using a structured sparsity regularization scheme, making it more robust to errors and outliers.

3 Background

In the following section, we first formulate the problem for 2D-PCA,then reformulate it for the 2D-PCA with structured sparsity regularization method. Finally, we outline the algorithm described by the authors to solve the optimization problem for the latter formulation.

3.1 Problem Formulation

Though the proposed algorithm can be used for a variety of applications involving two-dimensional signals, we have presented the problem and solution in terms of image and video processing given its obvious use case. Let $\mathcal{Y} = \{\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_N\} \subset \mathbb{R}^{m \times n}$ be the set of N images or frames of a video, each with m rows and n columns. First, the images are scaled such that the the pixel values line in the range $[0, 1]$. Then, the mean image $\bar{\boldsymbol{Y}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{Y}_i$ is subtracted from each image. With these preprocessed images, the 2D-PCA problem is written as

$$\begin{aligned} \min_{\boldsymbol{U},\boldsymbol{V},\{\boldsymbol{\Sigma}_i\}} \quad & \frac{1}{2N}\sum_{i=1}^{N} \left\|\boldsymbol{Y}_i - \boldsymbol{U}\boldsymbol{\Sigma}_i\boldsymbol{V}^T\right\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_r,\ \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_c \end{aligned} \tag{1}$$

where $\boldsymbol{U} \in \mathbb{R}^{m \times r}$, $\boldsymbol{\Sigma}_i \in \mathbb{R}^{r \times c}$, and $\boldsymbol{V} \in \mathbb{R}^{n \times c}$ for $r \leq m$ and $c \leq n$ and the optimal solution seeks to minimize the reconstruction error between each image $\boldsymbol{Y}_i$ and its approximation $\boldsymbol{U}\boldsymbol{\Sigma}_i\boldsymbol{V}^T$. The minimization problem seeks to find a bi-directional decomposition of each image $\boldsymbol{Y}_i$ using the bases $\boldsymbol{U}$ and $\boldsymbol{V}$ where $\boldsymbol{\Sigma}_i$ is the low-dimensional representation of $\boldsymbol{Y}_i$ by identifying spatial redundancies in the data. This objective function based on the Frobenius norm is optimal only when the residuals can be modeled as densely distributed Gaussian noise (an assumption that is hardly true for real data) and is sensitive to outliers. The objective function for 2D-PCA can be reformulated to better handle outliers and corrupted data:

$$\begin{aligned} \min_{\boldsymbol{U},\boldsymbol{V},\{\boldsymbol{\Sigma}_i,\boldsymbol{E}_i\}} \quad & \frac{1}{2N}\sum_{i=1}^{N} \left\|\boldsymbol{Y}_i - \left(\boldsymbol{U}\boldsymbol{\Sigma}_i\boldsymbol{V}^T + \boldsymbol{E}_i\right)\right\|_F^2 + \lambda\|\boldsymbol{E}_i\|_1 + \sum_{j=1}^{J}\beta_j\left\|\boldsymbol{E}_{i,\boldsymbol{\Omega}_j}\right\|_2 \\ \text{s.t.} \quad & \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_r,\ \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_c \end{aligned} \tag{2}$$

where $\boldsymbol{E}_i \in \mathbb{R}^{m \times n}$ approximates the outliers and corrupted data for each image $\boldsymbol{Y}_i$, $\boldsymbol{\Omega}_j$ denotes a subset of matrix indices for the j-th group of the structured sparse matrix $\boldsymbol{E}_i$, and elements of $\boldsymbol{E}_{i,\boldsymbol{\Omega}_j}$ at indices in $\boldsymbol{\Omega}_j$ are equal to the corresponding elements in $\boldsymbol{E}_i$ and zero otherwise. In contrast to Eq. (1), Eq. (2) seeks the best approximation $\boldsymbol{U}\boldsymbol{\Sigma}_i\boldsymbol{V}^T + \boldsymbol{E}_i$ of $\boldsymbol{Y}_i$ with two regularization terms. The l_1-norm term seeks to promote sparsity in the error term $\boldsymbol{E}_i$. The second regularization term is a result of the work from [8] and seeks to identify J spatial clusters of outliers amongst the N images.

3.2 Two-Stage Alternating Minimization Algorithm

As Eq. (2) is neither convex nor smooth, the authors propose a two-stage alternating minimization algorithm to find the optimal solution. The first step of the algorithm identifies the optimal projection

matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ for the images in $\mathcal{Y}$. If we assume $\boldsymbol{\Sigma}_i$ and $\boldsymbol{E}_i$ to be constant, then Eq. (2) can be rewritten in terms of a matrix trace and the projection matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ can be found by solving the following optimization problem

$$\begin{aligned} \max_{\boldsymbol{U},\boldsymbol{V}} \quad & \frac{1}{N}\sum_{i=1}^{N} tr[(\boldsymbol{Y}_i - \boldsymbol{E}_i)^T \boldsymbol{U}\boldsymbol{U}^T(\boldsymbol{Y}_i - \boldsymbol{E}_i)\boldsymbol{V}\boldsymbol{V}^T] \\ \text{s.t.} \quad & \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_r,\ \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_c \end{aligned} \tag{3}$$

To solve this problem and find the values for $\boldsymbol{U}$ and $\boldsymbol{V}$ we iteratively maximize the covariance matrices in two directions. Specifically, to find $\boldsymbol{U}$ we fix $\boldsymbol{V}$ and solve the following reformulation of Eq. (3)

$$\begin{aligned} \max_{\boldsymbol{U}} \quad & \text{tr}[\boldsymbol{U}^T C_v \boldsymbol{U}] \\ \text{s.t.} \quad & \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_r \end{aligned} \tag{4}$$

where $\boldsymbol{C}_v = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{Y}_i - \boldsymbol{E}_i)\boldsymbol{V}\boldsymbol{V}^T(\boldsymbol{Y}_i - \boldsymbol{E}_i)^T$. $\boldsymbol{V}$ is similarly obtained by fixing $\boldsymbol{E}$ which results in the same formulation as Eq. (5) with the positions of $\boldsymbol{U}$ and $\boldsymbol{V}$ being swapped in the equations, which can be expressed as:

$$\begin{aligned} \max_{\boldsymbol{V}} \quad & \text{tr}[\boldsymbol{V}^T C_u \boldsymbol{V}] \\ \text{s.t.} \quad & \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_c \\ & \boldsymbol{C}_u = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{Y}_i - \boldsymbol{E}_i)\boldsymbol{U}\boldsymbol{U}^T(\boldsymbol{Y}_i - \boldsymbol{E}_i)^T \end{aligned} \tag{5}$$

Then we need to use the structured sparsity to operated on Eq. (2) and the following can be get:

$$\min_{\boldsymbol{E}_i} \quad \frac{1}{N}||\boldsymbol{M}_i - \boldsymbol{E}_i||_F^2 + \lambda||\boldsymbol{E}_i||_1 + \sum_{j=1}^{J}\beta_j||\boldsymbol{E}_{i\Omega_j}||_p \tag{6}$$

where $\boldsymbol{M}_i = \boldsymbol{Y}_i - \boldsymbol{U}\Sigma_i\boldsymbol{V}^T$ is always constant, if we realize the column vectors of $\boldsymbol{E}_i$ as $\mathbf{e} \in \mathbb{R}^{mn}$, then the solution of Eq. (6) can be expressed as:

$$C = \{\mathbf{e} \in \mathbb{R}^{mn} \mid ||\mathbf{e}||_1 + \sum_{j=1}^{J}\omega_j||\mathbf{e}_{\Omega_j}||_2 \leq s\} \tag{7}$$

For the Feature Matrix Extraction, when $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{E}_i$ is considered to be constant, we need to set the partial derivative of Eq. (2) as zero to get the expression of Σ_i which is:

$$\Sigma_i = \boldsymbol{U}^T(\boldsymbol{Y}_i - \boldsymbol{E}_i)\boldsymbol{V} \tag{8}$$

In the second step, these projection matrices are used to extract the feature matrix $\boldsymbol{\Sigma}_i$ and structured outlier matrix $\boldsymbol{E}_i$ for each image $\boldsymbol{Y}_i$. Pseudocode for each stage of the minimization is presented in Algorithms 1 and 2.

Algorithm 1: Iterative Bi-Directional Decomposition

Input: $\{\boldsymbol{Y}_i\}_{i=1}^N, \{\boldsymbol{E}_i\}_{i=1}^N, \boldsymbol{U}^{(0)}, \boldsymbol{V}^{(0)}, t = 1$
Output: $\boldsymbol{U}^*, \boldsymbol{V}^*$

while *not converged* **do**
- $\boldsymbol{C}_v \leftarrow \frac{1}{N}(\boldsymbol{Y}_i - \boldsymbol{E}_i)\boldsymbol{V}^{(t-1)}\boldsymbol{V}^{(t-1)^T}(\boldsymbol{Y}_i - \boldsymbol{E}_i)^T$;
- $\boldsymbol{U}^{(t)} \leftarrow [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r]$;
- $\boldsymbol{C}_u \leftarrow \frac{1}{N}(\boldsymbol{Y}_i - \boldsymbol{E}_i)\boldsymbol{U}^{(t-1)}\boldsymbol{U}^{(t-1)^T}(\boldsymbol{Y}_i - \boldsymbol{E}_i)^T$;
- $\boldsymbol{V}^{(t)} \leftarrow [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_c]$;
- $\boldsymbol{U}^* \leftarrow \boldsymbol{U}^{(t)}, \boldsymbol{V}^* \leftarrow \boldsymbol{V}^{(t)}$;
- $t \leftarrow t + 1$;

end
return $\boldsymbol{U}^*, \boldsymbol{V}^*$

Algorithm 2: Feature Matrix and Structured Outlier Extraction

Input: $\{\boldsymbol{Y}_i\}_{i=1}^N, \boldsymbol{E}_i = \boldsymbol{0}_{m\times n}, \boldsymbol{U}, \boldsymbol{V}, t = 1$
Output: $\boldsymbol{E}_i^*, \boldsymbol{\Sigma}_i^*$

while *not converged* **do**
- $\boldsymbol{\Sigma}_i^{(t)} \leftarrow \boldsymbol{U}^T(\boldsymbol{Y}_i - \boldsymbol{E}_i^{(t-1)})\boldsymbol{V}$;
- $\boldsymbol{E}_i^{(t)} \leftarrow \text{prox}_C(\boldsymbol{Y}_i - \boldsymbol{U}\boldsymbol{\Sigma}_i^{(t)}\boldsymbol{V}^T)$;
- $t \leftarrow t + 1$;

end
$\boldsymbol{E}_i^* \leftarrow \boldsymbol{E}_i^{(t)}, \boldsymbol{\Sigma}_i^* \leftarrow \boldsymbol{\Sigma}_i^{(t)}$;
return $\boldsymbol{E}_i^*, \boldsymbol{\Sigma}_i^*$

For a detailed derivation and description for each of the algorithms, please refer to the original paper [16].

4 Methods

In this section, we introduce our implementation and evaluation of Algorithms 1 and 2, the dataset, and the setup of the experiments that we ran. All experiments were conducted on a MacBook Pro with a 1.4 GHz Quad-Core Intel Core i5 processor with 16 GB of RAM. The original code is publicly available on GitHub[1].

4.1 Implementation

We re-implemented the algorithm in Python using only basic Python packages (NumPy [5] and SciPy [18]) based on the description provided by the authors. When solving the optimization problem using the proximal method in Algorithm 2, we used the `proximalFlat` method in the Python version of the SPAMS[2] package [11, 9]. The original paper did not specify the metric by which convergence was determined for both algorithms. Thus we chose to use the Frobenius norm of the difference between the previous $X^{(t-1)}$ and current $X^{(t)}$ steps as our convergence criterion (Equation 9)

$$\left\| X^{(t)} - X^{(t-1)} \right\|_F < \delta \tag{9}$$

where δ is a user-defined parameter.

4.2 Dataset

Once implemented, we evaluated the performance of the algorithm through a reconstruction experiment of corrupted images based on the Yale Face Database [4], which contains 165 grayscale 320-by-243 images of 15 individuals (11 per individual) under various lighting conditions and facial

[1] https://github.com/matsumotosan/SSR-R2D-PCA
[2] http://thoth.inrialpes.fr/people/mairal/spams/

configurations. We created partial occlusions in the images (set pixel values to zero) for a randomly chosen subset of the images. We chose to corrupt 20% of the images and occlusions were set to cover 10% of the image (Figure 1).

Figure 1: Examples of images from the Yale Face Database corrupted by occlusions.

4.3 Experiments

We evaluated the quality of the reconstructions on the corrupted datasets and compared them to the performance of scikit-learn's implementation of PCA [15]. We used PSNR, SSIM, and MSE as our quantitative metrics for the quality of reconstruction, each of which are evaluated in more detail below. We used the skimage implementation of these metrics in our code.

For PCA, we evaluated the reconstruction quality using {5,10,15,...,80} PCs. For structured sparsity regularized robust 2D-PCA (SSR-R2D-PCA), we evaluated the reconstruction quality for dimensionality reduction parameter $s = \{10, 15, 20\}$, where s is used to determine the number of columns in the projection matrices $(V^*)^{m\times\lfloor n/s\rfloor}$ and $(U^*)^{n\times\lfloor p/s\rfloor}$ where $\lfloor\cdot\rfloor$ is the floor function. The convergence parameters for `iterBDD` and `feature_outlier_extractor` were set to $\delta = 1\text{e}{-12}$ and $\delta = 1\text{e}{-3}$, respectively.

4.4 Evaluation Metrics

In order to accordingly evaluate the algorithms, three different evaluation methods were used. The methods include mean squared error (MSE), peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). In the following subtopics each are briefly introduced. All of these methods are used to compare two images, the original and the reconstructed image.

Mean Squared Error MSE computes the absolute error for each pixel of the images, squares it, and takes the average. Therefore, Eq. (10) can be used. Note that $\boldsymbol{Y}_i \in R^{m\times n}$ and $\hat{\boldsymbol{Y}}_i \in R^{m\times n}$.

$$MSE_i = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(\boldsymbol{Y}_i(i,j) - \hat{\boldsymbol{Y}}_i(i,j))^2 \tag{10}$$

where $\boldsymbol{Y}_i$ is the i-th original and $\hat{\boldsymbol{Y}}_i$ is the i-th reconstructed image.

Peak Signal-to-Noise Ratio PSNR describes the ratio between the maximum possible power of an signal and the power of corrupting noise. For the usage with images, the original image is seen as signal and the difference between the reconstructed and original image as noise. With Eq. (11) its value can be determined.

$$PSNR_i = 10\log_{10}\frac{255^2 \times mn}{||\boldsymbol{Y}_i - \hat{\boldsymbol{Y}}_i||_F^2} \tag{11}$$

where $\boldsymbol{Y}_i$ is the original and $\hat{\boldsymbol{Y}}_i$ the reconstructed image.

Structural Similarity Index Measure SSIM is a perception-based model. It considers image degradation and incorporates perceptual phenomena, for example luminance and contrast masking. It differs from MSE and PSNR as those approaches estimate the absolute error whereas SSIM takes object structures in visual scene into account. Human perception is thereby emphasized.

The SSIM can be calculated with the following equations. They have been taken from [19].

$$SSIM(x, y) = \frac{(2\mathbf{u}_x\mathbf{u}_y + C_1)(2\sigma_{xy} + C_2)}{(\mathbf{u}_x^2 + \mathbf{u}_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{12}$$

$\mathbf{u}_x$, $\mathbf{u}_y$ are the averages of x and y. σ_x^2, σ_y^2 are the variances of x and y accordingly. σ_{xy}^2 is the covariance between x and y. C_1 and C_2 can be calculated as $C_1 = (k_1 L)^2$ and $C_2 = (k_2 L)^2$ with L as the dynamic range of the pixel values and $k_1 = 0.01$, $k_2 = 0.03$.

5 Results

Figure 2 shows images reconstructed using PCA where the first image of each column is the original and the second is the corresponding corrupted image. The third column onward correspond to the reconstructed images using 5, 25, 45, and 65 PCs, respectively.

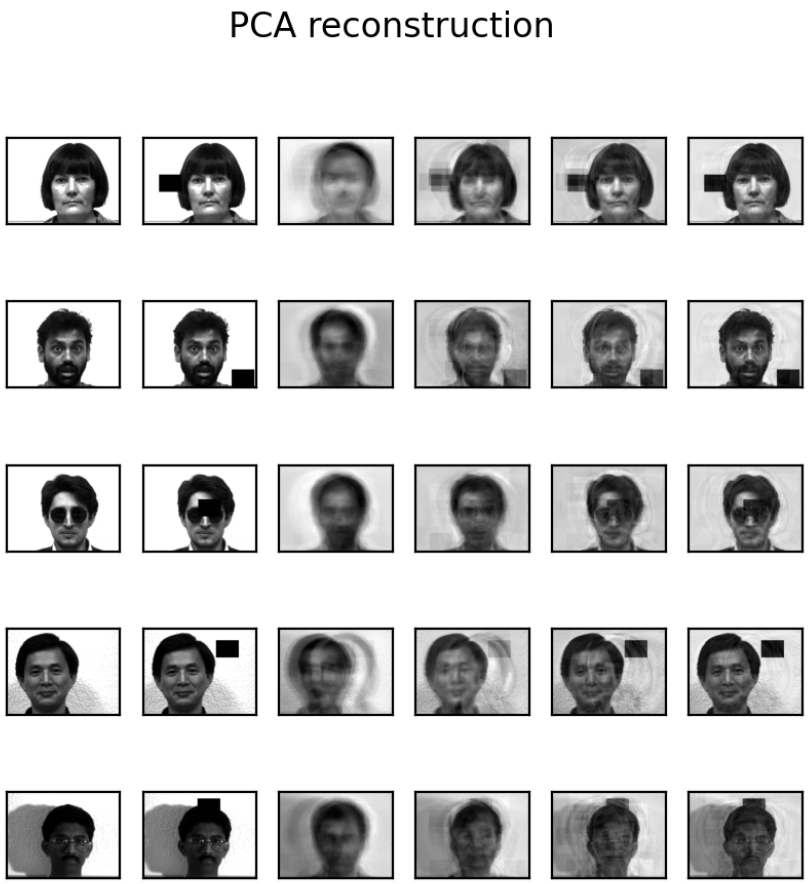

Figure 2: Reconstruction of Yale Faces using various numbers of PCs for PCA. Columns from left to right: original image, corrupted image, reconstructed images with 5, 25, 45, and 65 PCs.

As shown in Figure 2, PCA is unable to distinguish between the original image (portrait) and the noise/outlier (black box). Thus, though the reconstructed image of the person improves in quality with a larger number of PCs, so does the outlier.

Figure 3 shows the reconstructed images using SSR-R2D-PCA with downscaling parameters $s = \{10, 15, 20\}$ in the third, fourth, and fifth column, respectively. The first and second columns correspond to the original and corrupted images, respectively. The last column is the extracted structured outlier matrix E_i for each image.

The results for SSR-R2D-PCA are not as expected as the reconstructed images do not match the corresponding original images. Upon closer inspection, we observed that the reconstructed images

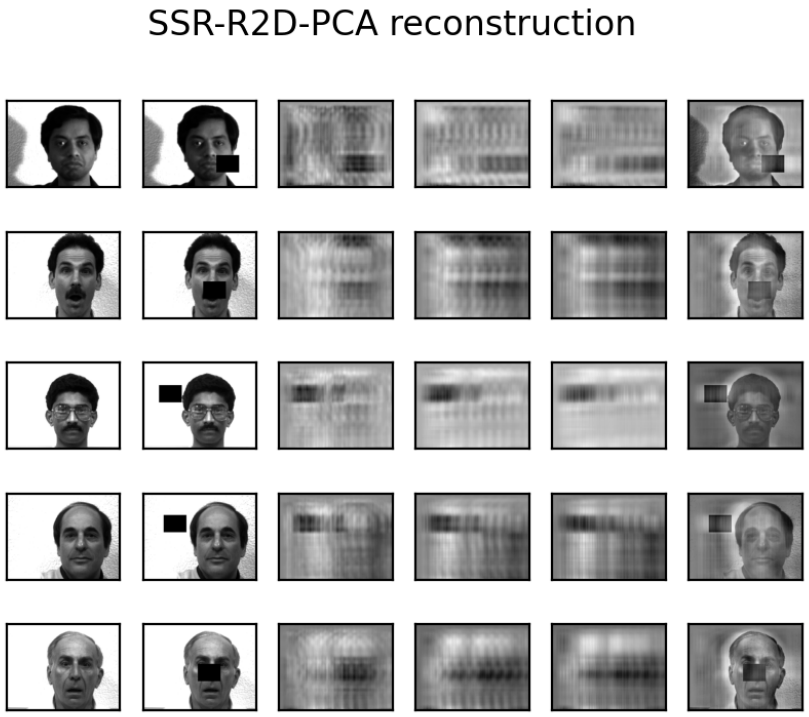

Figure 3: Reconstruction of images from the Yale Faces Database using various dimensionality reduction downscaling factors for SSR-R2D-PCA. Columns from left to right: original image, corrupted image, reconstructed images with $s = \{10, 15, 20\}$, and structured outlier matrix E_i.

faintly captured the original image, though it appeared to be repeated horizontally. The structured outlier matrices E_i not only captured the occlusion, but also the individual pictured in each image.

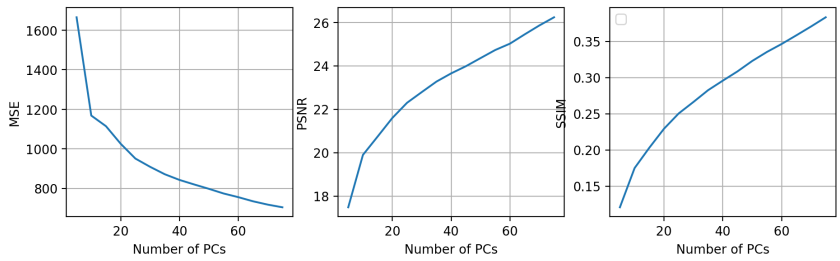

Figure 4: Metrics for PCA as a function of the number of PCs; left: MSE; middle: PSNR; right: SSIM.

The metrics evaluated for each method using various dimensionality reduction parameters are shown in Figures 4 and 5. The MSE, PSNR, and SSIM plots are as expected for PCA – the MSE decreases as the number of PCs increases and the PSNR and SSIM increases with the number of PCs. The metrics for SSR-R2D-PCA are not what was expected as the plots show that all three metrics worsen with an increasing number of PCs.

6 Discussion

We think that our inability to reconstruct the results as in the original paper is due to our use of the SPAMS package needed in solving the optimization problem presented in Step 3 of Algorithm 2, which is crucial to successfully identifying structured outliers in the images. The function requires us

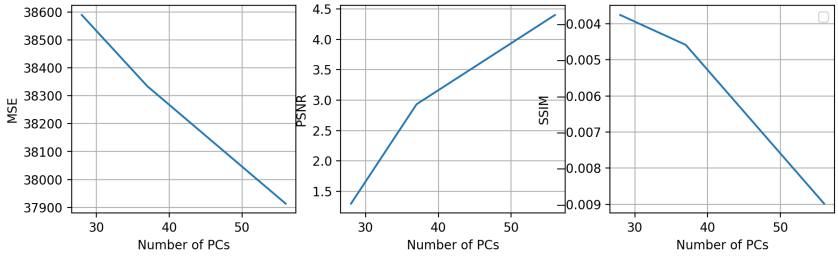

Figure 5: Metrics for SSR-R2D-PCA as a function of the number of PCs; left: MSE; middle: PSNR; right: SSIM.

to specify subgrids of pixels Ω_j as in Equation 7. However, we were not able to correctly specify the group structure based on the documentation[3] provided. Instead, in our implementation, we were only able to specify groups to be the sets of three consecutive, non-overlapping pixels of the vectorized form of the image. We believe that the banding artifact seen in the reconstructed images is a direct result of the group structure parameter that we set.

As a result, we hypothesize that the algorithm interpreted the image of the individual as an outlier in addition to the occlusion that was introduced. This is evident in the fact that each of the extracted structured outlier matrices E_i was able to capture the individual in each image and the occlusion whilst eliminating the background.

7 Conclusion

In this implementation study, we attempted to implement the SSR-R2D-PCA method presented in [16] in Python. We evaluated the performance of the algorithm to standard PCA in an image reconstruction study. The results showed that our implementation of the method was incorrect, which we hypothesize is mainly due to our incorrect use of the SPAMS package needed to a critical optimization problem presented in the algorithm.

References

[1] Ori Bryt and Michael Elad. "Compression of facial images using the K-SVD algorithm". In: *Journal of Visual Communication and Image Representation* 19.4 (May 2008), pp. 270–282. ISSN: 10473203. DOI: 10.1016/j.jvcir.2008.03.001.

[2] Emmanuel J. Candès et al. "Robust principal component analysis?" In: *Journal of the ACM*. Vol. 58. 3. May 2011. DOI: 10.1145/1970392.1970395.

[3] Ehsan Elhamifar and Rene Vidal. "Robust classification using structured sparse representation". In: *CVPR 2011*. IEEE, June 2011, pp. 1873–1879. ISBN: 978-1-4577-0394-2. DOI: 10.1109/CVPR.2011.5995664. URL: http://ieeexplore.ieee.org/document/5995664/.

[4] 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". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.6 (June 2001). ISSN: 01628828. DOI: 10.1109/34.927464.

[5] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020). ISSN: 0028-0836. DOI: 10.1038/s41586-020-2649-2.

[6] Jun He, Laura Balzano, and Arthur Szlam. "Incremental Gradient on the Grassmanian for Online Foreground and Background Separation in Subsampled Video". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1568–1575. ISBN: 9781467312288. DOI: 10.1109/CVPR.2012.6247848.

[7] Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. "Structured Sparse Principal Component Analysis". In: (Sept. 2009). URL: http://arxiv.org/abs/0909.1440.

[3] http://thoth.inrialpes.fr/people/mairal/spams/doc-python/html/doc$_spams006.html$

[8] Rodolphe Jenatton et al. “Proximal Methods for Hierarchical Sparse Coding”. In: *Journal of Machine Learning Research* 12 (July 2011), pp. 2297–2334.

[9] Rodolphe Jenatton et al. “Proximal Methods for Sparse Hierarchical Dictionary Learning”. In: (2010).

[10] Oleg Kuybeda et al. “A collaborative framework for 3D alignment and classification of heterogeneous subvolumes in cryo-electron tomography”. In: *Journal of Structural Biology* 181.2 (2013), pp. 116–127. ISSN: 1047-8477. DOI: https://doi.org/10.1016/j.jsb.2012.10.010. URL: https://www.sciencedirect.com/science/article/pii/S1047847712002924.

[11] Julien Mairal et al. “Convex and Network Flow Optimization for Structured Sparsity”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2681–2720.

[12] Julien Mairal et al. “Discriminative learned dictionaries for local image analysis”. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2008, pp. 1–8. ISBN: 978-1-4244-2242-5. DOI: 10.1109/CVPR.2008.4587652. URL: http://ieeexplore.ieee.org/document/4587652/.

[13] Xue Mei and Haibin Ling. “Robust visual tracking and vehicle classification via sparse representation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.11 (2011), pp. 2259–2272. ISSN: 01628828. DOI: 10.1109/TPAMI.2011.66.

[14] Lukas Meier, Sara Van De Geer, and Peter Bühlmann. “The group lasso for logistic regression”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.1 (Jan. 2008), pp. 53–71. ISSN: 13697412. DOI: 10.1111/j.1467-9868.2007.00627.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2007.00627.x.

[15] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12.85 (Oct. 2011), pp. 2825–2830.

[16] Yipeng Sun et al. “Robust 2D principal component analysis: A structured sparsity regularized approach”. In: *IEEE Transactions on Image Processing* 24.8 (Aug. 2015), pp. 2515–2526. ISSN: 10577149. DOI: 10.1109/TIP.2015.2419075.

[17] Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. “Subspace learning from image gradient orientations”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.12 (2012), pp. 2454–2466. ISSN: 01628828. DOI: 10.1109/TPAMI.2012.40.

[18] Pauli Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17.3 (Mar. 2020). ISSN: 1548-7091. DOI: 10.1038/s41592-019-0686-2.

[19] Z. Wang, E.P. Simoncelli, and A.C. Bovik. “Multiscale structural similarity for image quality assessment”. In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. IEEE, pp. 1398–1402. ISBN: 0-7803-8104-1. DOI: 10.1109/ACSSC.2003.1292216. URL: http://ieeexplore.ieee.org/document/1292216/.

[20] John Wright et al. *Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Matrices*. 2009.

[21] John Wright et al. “Sparse representation for computer vision and pattern recognition”. In: *Proceedings of the IEEE* 98.6 (2010), pp. 1031–1044. ISSN: 00189219. DOI: 10.1109/JPROC.2010.2044470.

[22] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. “Robust PCA via outlier pursuit”. In: *IEEE Transactions on Information Theory*. Vol. 58. 5. May 2012, pp. 3047–3064. DOI: 10.1109/TIT.2011.2173156.

[23] Ming Yuan and Yi Lin. “Model selection and estimation in regression with grouped variables”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1 (Feb. 2006), pp. 49–67. ISSN: 1369-7412. DOI: 10.1111/j.1467-9868.2005.00532.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2005.00532.x.