Performance Evaluation of Dimensionality Reduction Techniques on High Dimensional Data

Mandikal Vikram, Rakesh Pavan, Navadiya Dhruvikkumar Dineshbhai, Biju Mohan
National Institute of Technology Karnataka, Surathkal, India 575025
Email: {15it217.vikram, rp.171it154, navadiyadhruvikkumardineshbhai.171it225, biju}@nitk.edu.in

***Abstract*—With a large amount of data being generated each day, the task on analyzing and making inferences from data is becoming an increasingly challenging task. One of the major challenges is the curse of dimensionality which is dealt with by using several popular dimensionality reduction techniques such as ICA, PCA, NMF etc. In this work, we make a systematic performance evaluation of the efficiency and effectiveness of various dimensionality reduction techniques. We present a rigorous evaluation of various techniques benchmarked on real-world datasets. This work is intended to assist data science practitioners to select the most suitable dimensionality reduction technique based on the trade-off between the corresponding effectiveness and efficiency.**

***Keywords*—Dimensionality reduction, Performance evaluation, PCA, ICA, SVD, NMF**

I. Introduction

The ideal dimensionality of data is the expectaion value of its intrinsic dimensionality. Unfortunately, the data available in large quantities and wide varities such as digital images, audio, medical records, business records etc are seldom are often found with very large dimensionalities. This results in the curse of dimensionality and many undesirable properties of high-dimensional spaces. Dimensionality reduction techniques are often used to project such high dimensional data into their supposed intrinsic dimensionality.

Dimensionality reduction is defined as follows: Reducing the dimensions of data X from D to it's intrinsic dimensionality d where $d < D$ and often $d << D$. These seek techniques seek to retain the geometry of X in it's original dimensions as much as possible in the lower dimensions as well. Neither the geometry of the information complex nor the intrinsic dimensionality d of the dataset X is known. Consequently, dimensionality reduction is an ill-posed problem that must be fathomed by assuming certain properties of the data, (for example, its intrinsic dimensionality).

Although several theoretical properties of these dimensionality techniques there is a lack of benchmarking of these techniques on several real-world datasets. In this work we present a performance evaluation study of four dimensionality reduction techniques - Independent Component Analysis (ICA), Principal Component Analysis (PCA), Non-negative Matrix Factorization (NMF) and Singular Value Decomposition (SVD). The contribution of this work is summarized as follows:

- A novel approach for performance evaluation of dimensionality reduction techniques on varying size of the workload (data) and varying reduced dimensions.
- Measure the effectiveness by defining metrics on the reconstructed data.
- Measure the time taken for computing the parameters and transforming the data by varying the workload and the reduced dimensions.

II. Related Works

This section includes a brief description of the dimensionality reduction techniques we compare.

- **PCA:** This consists of the eigenvalue decomposition of the data covariance matrix, $\mathbf{E}\left\{XX^T\right\} = E\Lambda E^T$, where the eigenvectors are represented by the columns of matrix E and the corresponding eigenvalues are present in Λ. The data can be projected into a subspace whose basis are the top few eigenvectors (whose corresponding eigenvalues are the highest):

  $$X^{PCA} = E_k^T X$$

  where E_k contains top k eigenvectors. For data of dimensionality d, the cost of approximating the PCA parameters is given by $O\left(d^2N\right) + O\left(d^3\right)$, [1].
- **ICA:** ICA is based on an underlying assumption that data are linearly mixed by a set of separate independent sources. It demixes these signal sources according to their statistical independency measured by mutual information. Although this technique was not originally developed for dimensionality reduction, it can be used for dimensionality reduction as mentioned in [2].
- **NMF:** The objectives is to decompose the given data into two non-negative matrices such that the product of these two matrices approximately equals the original data matrix. Given X, the objective is to find non-negative matrix factors W and H such that $\boldsymbol{X} \approx \boldsymbol{WH}$.
- **SVD:** Singular Value Decomposition of data matrix X is given by $\boldsymbol{X} = USV^T$, where U contains the left singular vectors, V contains the right singular vectors and S contains the singular values. To obtain the reduced dimensional data, the data is projected onto the subspace defined by k left singular vectors corresponding to the k largest singular values:

  $$X^{SVD} = U_k^r X$$

  where U_k contains the k left singular values. The computational complexity for data with dimensionality

d and about c nonzero entries per column is of the order $O(dcN)$ [3].

Many other works such as [4], [5] and [6] provide a theoretical evaluation of the dimensionality reduction techniques, however, they lack appreciable practical comparison on real-world datasets. We address this shortcoming in our work where we propose an evaluation techniques with several metrics to compare the performance of these techniques on well-known datasets. [4] contains some comparison on practical datasets, however, they use 'continuity' of the low dimensional embeddings as their sole metric. We argue that continuity of the continuity of the embeddings is of no appreciable benefit compared to the richness of the information stored in the low-dimensional embeddings. We propose that this richness of the information can be measured by measure the goodness of the reconstructed data from the embeddings. We introduce several metrics for this and also give a comprehensive analysis of the variation of performance while changing the dimensionality of the embeddings.

III. Proposed Methodology

This involves computing the parameters of various techniques and measuring the associated efficiency and effectiveness metrics for increasing workloads. The efficiency metrics are as follows:

- Fit time - The total time taken to compute the parameters of a technique divided by the total number of data points in the train dataset.
- Transform time - The average time taken to transform one datapoint from the high dimensional space to lower dimensions.

The effectiveness of the techniques is measured by several metrics based on the similarity between the reconstructed datapoint and the original datapoint. Let X denote original datapoint, Y the datapoint with reduced dimensions, f be the dimensionality reduction technique. Hence $f(X) = Y$. The inverse transform f^{-1} projects the low dimensional data back to the higher dimensions, $f^{-1}(Y) = \hat{X}$, where $\hat{X}$ is the reconstructed datapoint. We compute the following metrics which are a function of $X, \hat{X}$ (PSNR and SSIM are computed only for image datasets):

- Mean squared error: It is simply the L2 distance between X and $\hat{X}$.

$$MSE(X, \hat{X}) = \|\mathbf{X} - \hat{\mathbf{X}}\|_2$$

- Peak-signal-to-noise ratio (PSNR): This is computed as follows,

$$PSNR(X, \hat{X}) = 10 \log_{10} \left(\frac{R^2}{MSE(X, \hat{X})} \right)$$

where R is the maximum fluctuation in the datatype of the image. PSNR approaches infinity as the MSE approaches zero; this shows that a higher PSNR implies that the two images are more similar. Similarly, a small value of the PSNR implies high numerical differences between images.

- Structural Similarity Index Measure (SSIM): This is computed as follows:

$$\text{SSIM}(X, \hat{X}) = l(X, \hat{X}) c(X, \hat{X}) s(X, \hat{X})$$

where

$$l(X, \hat{X}) = \frac{2\mu_X \mu_{\hat{X}} + C_1}{\mu_X^2 + \mu_{\hat{X}}^2 + C_1}$$

$$c(X, \hat{X}) = \frac{2\sigma_X \sigma_{\hat{X}} + C_2}{\sigma_X^2 + \sigma_{\hat{X}}^2 + C_2}$$

$$s(X, \hat{X}) = \frac{\sigma_X \hat{X} + C_3}{\sigma_X \hat{\sigma}_{\hat{X}} + C_3}$$

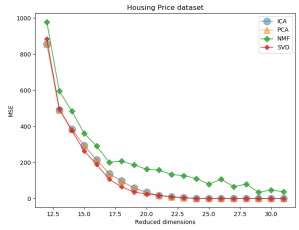
l is the lumninance comparison function, the maximal value of this function is 1 when both images are equal. Similarly, c is the contrast measuring function with a maximal value of 1 when the two images are equal. s measures the correlation coefficient between the two images. The positive constants C_1, C_2, C_3 are used to avoid zero denominator.

TABLE I
Results on housing price dataset

Metric	Dimensions	ICA	PCA	NMF	SVD
MSE	10	837.1	822.4	980.2	870.2
	15	260.2	261.7	386.8	242.5
	20	14.1	14.5	175.4	9.8
	30	0.04	0.04	32.1	0.03
log(Fit time)	10	-2	-2.5	-1.2	-2.4
	15	-2	-2.4	-1.1	-2.3
	20	-1.8	-2.3	-0.8	-2.2
	30	-1	-2.2	-0.6	-2.3
log(Transform time)	10	-3.5	-3.7	-2.5	-3.7
	15	-3.5	-3.6	-2.4	-3.8
	20	-3.5	-2.6	-2.0	-3.8
	30	-3.5	-3.6	-1.0	-3.8

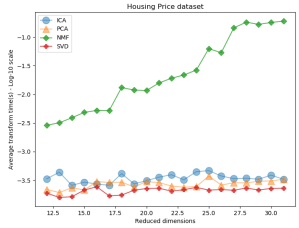
IV. Results

We first present results on the Housing price dataset [7], we considered only 38 numeric dimensions for our evaluation. The metrics are measured are presented in Table I and Figure 1. It can be observed that MSE reduces as we increase the number of reduced dimensions in al the techniques. Across all dimensions we observe that NMF has the highest MSE (worst performance) and also takes much longer time for both fitting and transforming. The time taken by NMF is orders of magnitude larger than the other techniques. It is also seen that order of time taken for transforming is fairly constant for all the techniques except NMF, where it increases as the reduced dimensions increases. It is seen that PCA and ICA have similar performance in terms of both MSE and transofrm time, however PCA does better in terms of fit-time (shorter fit time). It can also be seen that the order of fit-time of NMF is linearly increasing with the number of dimensions.



(a)

(b)



(c)

Fig. 1. Results on housing price dataset. (a) shows how the MSE varies as the reduced dimensions are increased. (b) and (c) show the fit time and transform time for varying reduced dimensions.

We then present results on standard image datasets with increasing number of dimensions. We present results on the MNIST dataset (784 dimensions), Tiny-Imagenet dataset (4906 dimensions) and the Caltech-256 dataset (59100 dimensions). We measure MSE, SSIM and PSNR for effectiveness and the fit time and transform for the efficiency.

The MNIST dataset[8] which is a popular dataset of handwritten digits with 60,000 training samples and 10,000 testing samples - with 784 dimensions. The results are presented is presented in table II and in Figure 2. It is observed that the MSE monotonically decreases as we increase the number of reduced dimensions. We also see that SSIM and PSNR monotonically increase as we increase the number of reduced dimensions - the reducing MSE and increasing PSNR and SSSIM shows that the reconstruction quality is improving as increase the number of reduced dimensions. We also see that NMF performs significantly worse than the other metrics with respect all the three effectiveness metrics. Similar to the housing price dataset, we observe that all though PCA and ICA have similar performance in-terms of effectiveness metrics, PCA is better

TABLE II
RESULTS ON MNIST DATASET

Metric	Dimensions	ICA	PCA	NMF	SVD
MSE	5	2.32E+9	2.32E+9	2.60E+9	2.35E+9
	10	1.79E+9	1.79E+9	2.29E+9	1.79E+9
	15	1.48E+9	1.48E+9	2.14E+9	1.48E+9
	20	1.27E+9	1.27E+9	2.05E+9	1.27E+9
	25	1.11E+9	1.11E+9	1.98E+9	1.11E+9
SSIM	5	0.36	0.36	0.30	0.35
	10	0.45	0.45	0.37	0.45
	15	0.51	0.51	0.40	0.50
	20	0.54	0.54	0.42	0.54
	25	0.57	0.57	0.43	0.57
PSNR	5	13.82	13.82	13.27	13.74
	10	14.95	14.95	13.87	14.94
	15	15.82	15.82	14.19	15.82
	20	16.49	16.49	14.37	16.50
	25	17.08	17.08	14.53	17.08
log(Fit time)	5	-2.37	-3.54	0.00	-3.66
	10	-2.42	-3.65	0.00	-3.37
	15	-2.31	-3.04	0.00	-3.64
	20	-2.21	-3.15	0.00	-3.12
	25	-2.40	-3.32	0.01	-3.15
log(Transform time)	5	-4.67	-4.31	0.00	-4.19
	10	-4.47	-4.84	0.00	-4.25
	15	-4.27	-4.81	0.00	-4.94
	20	-4.14	-4.81	0.00	-4.94
	25	-4.34	-4.39	0.00	-4.11

than ICA in terms of efficiency which can be seen in transform time and fit time values. NMF again takes a time which is higher in orders of magnitude for both fitting and transforming. Another observation is that the order of the transform time of all the four techniques remains constant as we increase the number of reduced dimensions.

The Tiny ImagenNet dataset[9] is a mini version of the famous ImageNet dataset. It contains 200 classes with each class having 500 train images, 50 validation and 50 test images. It is a standard dataset for benchmarking classification models. The results on this dataset are presented is presented in table III and in Figure 3. The obsevrvations are similar to that on MNIST except that we have now increased the range of the dimensions. We again see that MSE monotonically decreases while PSNR and SSIM monotonically increases as we increase the number of dimensions. It is more evident that the order of the time taken for fitting NMF is almost monotonically increasing as with the increasing embedding size. There is a significant gap in the performance in terms of MSE, SSIM and PSNR between NMF and the other techniques. This gap in performance is more evident in this dataset compared to the MNIST dataset due to the increase in the input dimensions. Another observation which is consistent with the previous two datasets is the similar effectiveness of both PCA and ICA, and the observation that ICA is more time consuming compared to PCA.

The Caltech-256 dataset[10] is a popular dataset which

TABLE III
RESULTS ON TINY-IMAGENET DATASET

Metric	Dimensions	ICA	PCA	NMF	SVD
MSE	5	3.92E+9	3.92E+9	4.41E+9	3.93E+9
	15	2.88E+9	2.88E+9	4.07E+9	2.87E+9
	25	2.57E+9	2.57E+9	3.99E+9	2.57E+9
	35	2.37E+9	2.38E+9	3.95E+9	2.38E+9
	45	2.24E+9	2.24E+9	3.93E+9	2.24E+9
	55	2.14E+9	2.14E+9	3.87E+9	2.14E+9
SSIM	5	0.03	0.03	0.01	0.03
	15	0.08	0.08	0.04	0.08
	25	0.11	0.11	0.04	0.11
	35	0.14	0.14	0.05	0.14
	45	0.16	0.16	0.05	0.16
	55	0.17	0.17	0.05	0.17
PSNR	5	13.49	13.49	12.93	13.49
	15	14.80	14.80	13.28	14.82
	25	15.36	15.36	13.35	15.36
	35	15.72	15.72	13.40	15.72
	45	15.98	15.98	13.42	15.99
	55	16.20	16.19	13.49	16.19
log(Fit time)	5	-2.06	-2.87	0.00	-3.04
	15	-1.95	-2.80	0.01	-2.90
	25	-2.12	-2.75	0.01	-2.86
	35	-2.07	-2.45	0.02	-2.66
	45	-2.06	-2.53	0.04	-2.53
	55	-2.05	-2.38	0.05	-2.69
log(Transform time)	5	-3.19	-4.15	0.00	-4.31
	15	-3.47	-4.03	0.00	-4.01
	25	-3.48	-3.98	0.00	-3.97
	35	-3.72	-3.47	0.00	-4.13
	45	-3.79	-3.88	0.00	-3.93
	55	-3.34	-3.78	0.00	-3.84

TABLE IV
RESULTS ON CALTECH-256 DATASET

Metric	Dimensions	ICA	PCA	NMF	SVD
MSE	5	7.84E+9	7.84E+9	8.75E+9	7.85E+9
	10	6.91E+9	6.91E+9	8.18E+9	6.84E+9
	20	6.10E+9	6.11E+9	8.14E+9	6.02E+9
	40	5.49E+9	5.49E+9	8.11E+9	5.42E+9
	60	5.11E+9	5.11E+9	8.08E+9	5.05E+9
	80	4.90E+9	4.90E+9	8.23E+9	4.84E+9
	95	4.78E+9	4.78E+9	8.22E+9	4.72E+9
SSIM	5	0.03	0.03	0.03	0.03
	10	0.04	0.04	0.04	0.04
	20	0.05	0.05	0.04	0.05
	40	0.06	0.06	0.04	0.06
	60	0.07	0.07	0.04	0.07
	80	0.08	0.08	0.04	0.08
	95	0.08	0.08	0.04	0.09
PSNR	5	15.41	15.41	14.95	15.38
	10	15.95	15.95	15.23	15.99
	20	16.50	16.49	15.24	16.55
	40	16.96	16.96	15.26	17.01
	60	17.27	17.27	15.27	17.31
	80	17.45	17.45	15.20	17.50
	95	17.57	17.57	15.20	17.62
log(Fit time)	5	-0.92	-1.09	0.12	-1.25
	10	-0.93	-1.21	0.26	-1.14
	20	-0.83	-1.08	0.86	-0.96
	40	-0.66	-0.80	3.17	-0.75
	60	-0.43	-0.66	6.79	-0.62
	80	-0.40	-0.44	13.79	-0.53
	95	-0.16	-0.42	19.60	-0.43
log(Transform time)	5	-2.38	-2.84	0.02	-2.58
	10	-2.21	-2.42	0.06	-2.69
	20	-2.18	-2.59	0.08	-2.08
	40	-2.00	-2.08	0.24	-1.88
	60	-1.73	-1.89	0.99	-2.06
	80	-1.61	-1.53	0.98	-1.73
	95	-1.51	-1.45	3.60	-1.77

contains 256 object categories. We particularly selected the car images from Caltech-256 dataset to investigate how the techniques perform on higher dimensional data (59100 dimensions) with only a few examples. This would show how techniques perform when there is a shortage of available data. The results are presented is presented in table IV and in Figure 4. It is again seen that MSE monotonically decreases as we increase the number of reduced dimensions. We also see that SSIM and PSNR monotonically increase as we increase the number of reduced dimensions for PCA, ICA and SVD, however, the increase stops after a point in NMF. It can be obseveved that the gap in the fit time of NMF and other datasets is much lager in this dataset compared to the previous datasets due to the increased input dimensions and reduced data samples. Another observation we can see is that the transform time of the four techniques other than NMF is of a constant order as we increase the embedding dimensions - the order of transform time for NMF is almost monotonically increasing.

From Figures 1, 2, 3 and 4 we can see that the quality of the images increase as we increase the number of dimensions, however this costs extra time for fitting and transforming. We observe that the time taken by NMF is consistently higher than the other techniques by a few orders of magnitude and also results in lower quality images. Although PCA and ICA often result in the same image quality, PCA is generally slightly faster than ICA. We also observe that SVD gives the best performance both in terms of speed and reconstruction quality.

V. CONCLUSION

In this work, we present evaluation study of the dimensionality reduction techniques PCA, ICA, SVD and NMF on real-world datasets. This is a rigorous practical evaluation of the fundamental dimensionality reduction techniques which differs from any of the previous works which are theoretically motivated. We introduce several novel techniques to assess the quality of the low-dimensional embeddings. We measure the effectivness by using metrics which assess how similar is the re-constructed data (from the embeddings) when compared to the original data. We measure the effectiveness by measuring the time taken to compute the parameters and the time taken to transform the data into the embedding space. We vary the dimensionality of the workload (the input data) and the embedding dimensionality while measuring the various metrics for effectiveness and efficiency. We intend to extend

this performance evaluation approach to other deep-learning based techniques such as auto-encoders.

REFERENCES

[1] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2012, vol. 3.

[2] J. Wang and C.-I. Chang, "Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis," *IEEE transactions on geoscience and remote sensing*, vol. 44, no. 6, pp. 1586–1600, 2006.

[3] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent semantic indexing: A probabilistic analysis," *Journal of Computer and System Sciences*, vol. 61, no. 2, pp. 217–235, 2000.

[4] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: a comparative," *J Mach Learn Res*, vol. 10, pp. 66–71, 2009.

[5] C. O. S. Sorzano, J. Vargas, and A. P. Montano, "A survey of dimensionality reduction techniques," *arXiv preprint arXiv:1403.2877*, 2014.

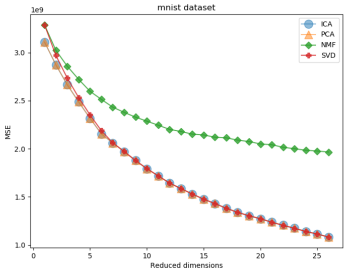
[6] I. K. Fodor, "A survey of dimension reduction techniques," Lawrence Livermore National Lab., CA (US), Tech. Rep., 2002.

[7] B. Park and J. K. Bae, "Using machine learning algorithms for housing price prediction," *Expert Syst. Appl.*, vol. 42, no. 6, pp. 2928–2934, Apr. 2015. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2014.11.040

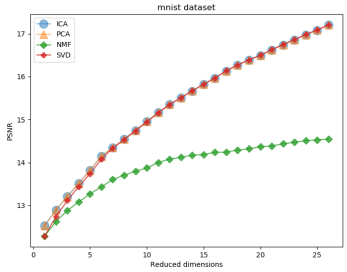
[8] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[9] P. Chrabaszcz, I. Loshchilov, and F. Hutter, "A downsampled variant of imagenet as an alternative to the cifar datasets," *arXiv preprint arXiv:1707.08819*, 2017.

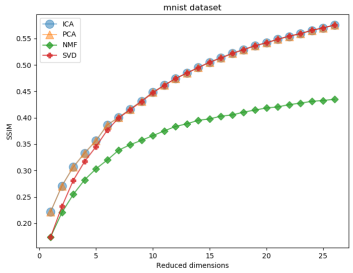
[10] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.



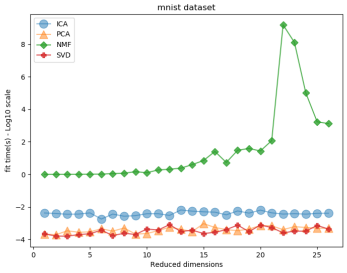
(a)



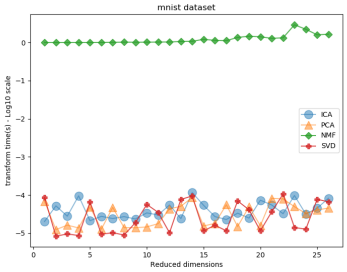
(b)



(c)

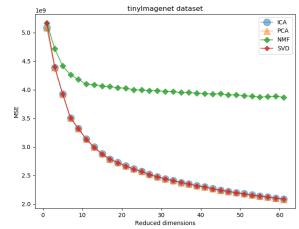


(d)

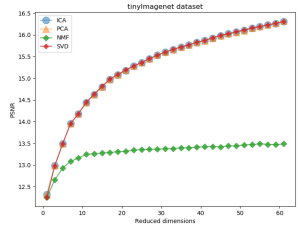


(e)

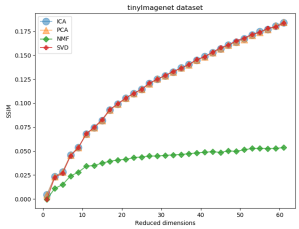
Fig. 2. Results on the MNIST dataset. (a) shows how the MSE varies as the reduced dimensions are increased. (b) and (c) show the PSNR and SSIM for varying reduced dimensions. (d) and (e) show the fit time and transform time for varying reduced dimensions.



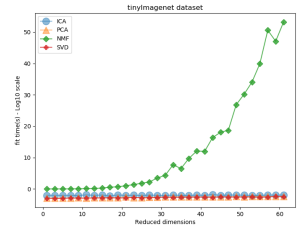
(a)



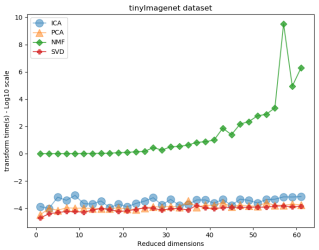
(b)



(c)

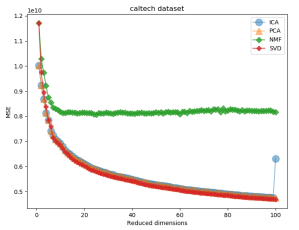


(d)

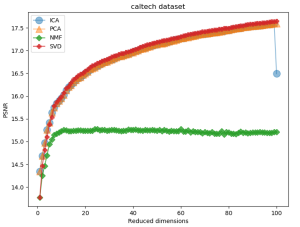


(e)

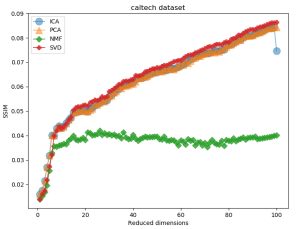
Fig. 3. Results on the TinyImageNet dataset. (a) shows how the MSE varies as the reduced dimensions are increased. (b) and (c) show the PSNR and SSIM for varying reduced dimensions. (d) and (e) show the fit time and transform time for varying reduced dimensions.



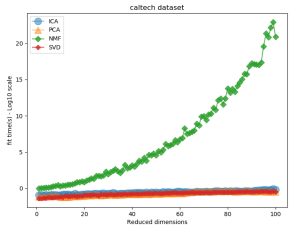
(a)



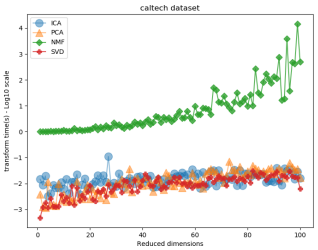
(b)



(c)



(d)



(e)

Fig. 4. Results on the car class images of Caltech256 dataset. (a) shows how the MSE varies as the reduced dimensions are increased. (b) and (c) show the PSNR and SSIM for varying reduced dimensions. (d) and (e) show the fit time and transform time for varying reduced dimensions.