

COMPRESSIVE SENSING WITH TENSORIZED AUTOENCODER

Rakib Hyder and M. Salman Asif*

University of California Riverside

ABSTRACT

Deep networks can be trained to map images into a low-dimensional latent space. In many cases, different images in a collection are articulated versions of one another; for example, same object with different lighting, background, or pose. Furthermore, in many cases, parts of images can be corrupted by noise or missing entries. In this paper, our goal is to recover images without access to the ground-truth (clean) images using the articulations as structural prior of the data. Such recovery problems fall under the domain of compressive sensing. We propose to learn autoencoder with tensor ring factorization on the the embedding space to impose structural constraints on the data. In particular, we use a tensor ring structure in the bottleneck layer of the autoencoder that utilizes the soft labels of the structured dataset. We empirically demonstrate the effectiveness of the proposed approach for inpainting and denoising applications. The resulting method achieves better reconstruction quality compared to other generative prior-based self-supervised recovery approaches for compressive sensing.

1. INTRODUCTION

Low-rank tensor factorization is a powerful tool to represent multi-dimensional and multi-modal data using a small number of low-dimensional factors (cores) [1]. Tensor factorization has also been recently used for compressing data and neural network parameters [2, 3]. Deep autoencoders and generative models, such as generative adversarial networks (GAN) [4], variational autoencoders (VAE) [5], and generative latent optimization (GLO) [6], also provide an excellent mechanism to learn low-dimensional representation of data.

In this paper, we combine tensor factorization with an autoencoder to recover a collection of articulated images from their corrupted or compressive measurements. Such articulated images often arise in surveillance and multi-view sensing applications [7, 8, 9]. Images with different imperfections (or their indirect measurements) can be modeled as

$$y_i = \mathbf{A}_i x_i + \eta_i, \quad (1)$$

where x_i denotes the i^{th} image, y_i denotes the observed measurements, \mathbf{A}_i denotes the corresponding measurement

matrix (corruption model), and η_i denotes the corresponding measurement noise.

Our main goal is to recover the structured images $\{x_i\}$ from the available measurements without access to the ground-truth (clean) images. To achieve this goal, we use the known image articulations as structural prior of the data. In particular, we learn an encoder that maps every measurement to a latent space. We represent the latent codes as a low-rank tensor, where different factors represent different image articulations. We learn a decoder that maps the low-rank tensor to the images. We present several experiments to demonstrate that our proposed method outperforms other self-supervised methods that use low-rank tensors or generative priors.

1.1. Related Work

Tensor factorization has been a popular method for multidimensional data and complex network compression [10, 11]. Different tensor factors can be used to represents different attributes of the data and potentially generate novel/missing data [12, 13].

We are using the strength of tensor factorization to compress the latent space of the deterministic autoencoder using explicit low-rank constraints in order to use it as a generative prior. We are also using it to factorize the representation of structured set of images in the latent space to achieve better recovery performance.

Compressive sensing refers to a broad class of problems in which we aim to recover a signal from a small number of measurements [14, 15, 16]. The canonical problem in (1) can be underdetermined, and we need to use some prior knowledge about the signal structure. Classical signal priors exploit sparse and low-rank structures in images and videos for their reconstruction [17, 18, 19, 20]. However, the natural images exhibits far richer nonlinear structures than sparsity alone. We focus on a newly emerging family of data-driven representation methods based on generative models that are either learned from training data or measurements [21, 22, 23]. A number of methods optimize generator network weights while keeping the latent code fixed at a random value [22, 24]. Both DIP [23] and deep decoder [22] update the network parameters to generate a given image; therefore, the generator can reconstruct wide range of images.

*This work was supported in part by ONR N00014-19-1-2264 and AFOSR FA9550-21-1-0330 grants.

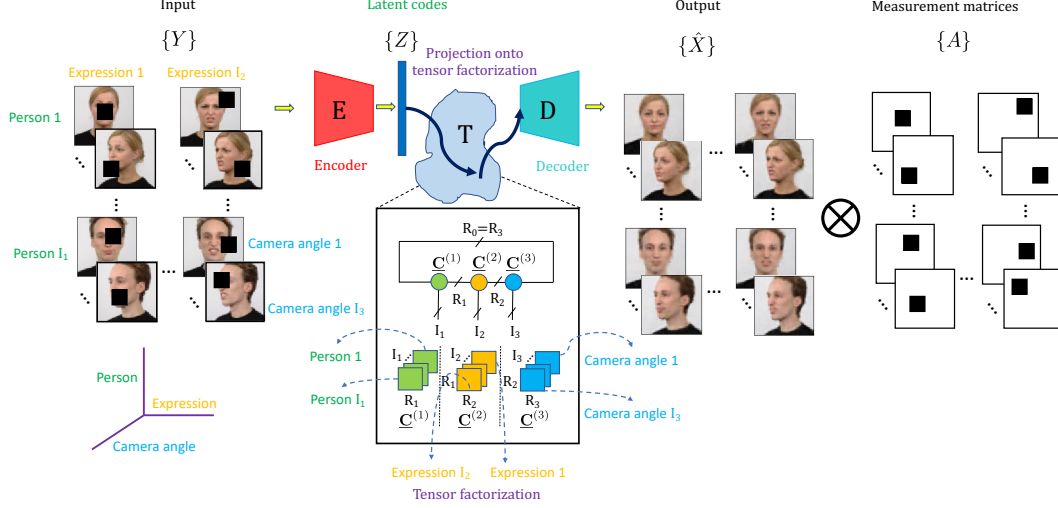


Fig. 1. General overview of our proposed tensor ring factorized autoencoder. We map a set of measurements $\{Y\}$ to latent codes $\{Z\}$ using an encoder E . We then perform tensor factorization on the latent space codes using tensor factorization (shown as T block). Finally, we pass the factorized representation through the decoder D to generate target images \hat{X} .

2. TECHNICAL DETAILS

Given corrupted or compressed measurements $\{y_i\}$ and measurement matrices $\{A_i\}$, our aim is to estimate the true image collection $\{x_i\}$. We learn a deterministic autoencoder by solving the following constrained optimization problem:

$$\min_{\theta, \gamma} \sum_{i=1}^N \|y_i - A_i D(E(y_i; \theta); \gamma)\| \text{ s.t. } E(y_i; \theta) \in \mathbb{T}, \quad (2)$$

where $E(\cdot; \theta)$ denotes an encoder, $D(\cdot; \gamma)$ denotes a decoder, \mathbb{T} represents the tensor-factorized latent space, and N denotes the total number of samples in the target set. For any measurement y_i , we denote the latent code representation as $z_i = E(y_i; \theta) \in \mathbb{R}^d$.

Tensor factorization can represent multi-dimensional and multi-modal data using a small number of low-dimensional factors. Instead of applying the tensor factorization directly on the image/signal space, we factorize the low-dimensional latent space. We seek two main goals with such factorization: 1) limit the degrees of freedom for the latent space and 2) utilize the structural similarity of the dataset for the self supervised compressive sensing.

We denote the N latent codes of our entire target set as $\{z_1, \dots, z_N\}$. For a structured dataset, these latent codes can be factored into K different attributes, each of which has I_k variants for $k \in \{1, 2, \dots, K\}$; therefore, we can write $N = I_1 \times I_2 \times \dots \times I_K$. We denote the latent code tensor with all the z_i as \mathcal{Z} , which is an $I_1 \times I_2 \times \dots \times I_K \times d$ tensor, and $\mathcal{Z}(i_1, \dots, i_K)$ denotes one of the d -dimensional latent codes and $\mathcal{Z}(i_1, \dots, i_K, i_{K+1})$ denotes one of the entries of a d -dimensional latent code.

We use tensor ring factorization in this paper. Brief description for tensor ring factorization is presented below. For detailed discussion, we refer the readers to [25].

A *tensor ring* (TR) decomposition can represent a latent tensor \mathcal{Z} using $K + 1$ different 3-order tensor cores: $\underline{C}^{(1)}, \dots, \underline{C}^{(K+1)}$ representing K different attributes and d dimensional code, where $\underline{C}^{(k)} \in \mathbb{R}^{R_{k-1} \times I_k \times R_k}$ and (R_1, \dots, R_{K+1}) denotes the multilinear rank with $R_0 = R_{K+1}$. All the entries in \mathcal{Z} can be represented as

$$\mathcal{Z}(i_1, \dots, i_{K+1}) = \sum_{R_1, \dots, R_{K+1}} \prod_{j=1}^{K+1} \underline{C}^{(j)}(R_{j-1}, i_j, R_j), \quad (3)$$

where $\underline{C}^{(k)}(:, i_k, :)$ denotes i_k th slice of $\underline{C}^{(k)}$ that is an $R_{k-1} \times R_k$ matrix and the trace operation sums up all the diagonal entries.

The total number of elements in \mathcal{Z} is $d \prod_{k=1}^K I_k$. The total number of parameters in TR factorization reduces to $\sum_{k=1}^K I_k R_{k-1} R_k + d R_K R_0$ with $R_0 = R_{K+1}$. If we set all the $R_k = R$, then the total number of parameters in TR factorization becomes $R^2(\sum_k I_k + d)$, which is significantly less than $d \prod_k I_k$ in \mathcal{Z} .

We modify the optimization problem in (2) to use the following loss function:

$$\begin{aligned} \text{Loss} = & \sum_{i=1}^N \|E(y_i; \theta) - \mathcal{Z}(i_1, \dots, i_K)\|^2 \\ & + \lambda_1 \|A_i D(E(y_i; \theta); \gamma) - y_i\|^2 \\ & + \lambda_2 \|A_i D(\mathcal{Z}(i_1, \dots, i_K); \gamma) - y_i\|^2. \end{aligned} \quad (4)$$

The three terms in the loss function in (4) are targeted to minimize the mismatch between encoder output and factorization,

Algorithm 1 Learning Tensor Factorization using Autoencoder from Corrupted Data

Input: Measurements $\{y_i\}$, measurement model \mathbf{A}_i , and attribute label of the data.

Initialize encoder and decoder weights and tensor cores randomly

for $m = 1, 2, \dots, M$ **do** $\triangleright M$ steps or until convergence
 Calculate the loss function in (4).

 Calculate gradients of loss w.r.t. training parameters in \mathbb{T} , $E(\cdot; \theta)$ and $D(\cdot; \gamma)$ via backpropagation.

 Update parameters using gradient descent.

end for

Intermediate Output: Optimized \mathbb{T} , $E(\cdot; \theta)$ and $D(\cdot; \gamma)$.

Use optimized \mathbb{T} and $D(\cdot; \gamma)$ to estimate the unknown signal as $\hat{x}_i = D(z_i; \gamma)$ where $z_i \in \mathbb{T}$.

Output: $\{x_i\}$

encoder-decoder measurement loss and factorization-decoder measurement loss respectively. λ_1 and λ_2 are weights for different loss terms. The first term of the total loss in (4) measures the mismatch between encoder output and factorization. The second term of the total loss in (4) evaluates how well the encoder output performs in terms of reconstruction. The third term in (4) measures how well the decoder performs when given the latent codes formed by the tensor factors. Note that if the first term were perfectly zero, the third term would not be necessary. Even though the output from the encoder and the output from the tensor factors are very close, they may provide very different realizations when passed through decoder depending on the direction of mismatch. We kept an extra term to make sure that the latent representation from the tensor factorization also generates as good images as encoder output does. We present a pseudocode for the recovery algorithm in Algorithm 1.

3. EXPERIMENTS AND RESULTS

Dataset: We used Small NORB [26], RaFD [27] and 3dShapes [28] datasets in our experiments. In these datasets, images of different attribute variation is available. We select 25 toys with 3 lighting conditions, 3 elevations and 9 azimuth angles (2025 images) from Small NORB dataset. We select 15 persons with 5 camera angles, 8 expressions and 3 eye gazing (1800 images) from RaFD. We selected 4 object shapes, 5 floor colors and 5 floor colors at 8 object scales (800 images) from 3dShapes dataset.

Setup: In our experiments, we used a fully convolutional autoencoder that maps each image to a latent code $z \in \mathbb{R}^{16 \times 4 \times 4}$, which sets $d = 256$ in our experiments. Our encoder consists of four convolutional layers (32,64,128,16 filters) and decoder consists of four transpose convolutional layers (256, 128,64,3 (rgb) /1 (grayscale) filters) each with 3×3 filters with stride=2 followed by ReLU activation except

for the last layer (that uses Sigmoid instead of ReLU). We use low-rank tensor ring factorization in the latent space. We use the same rank for all the cores of tensor ring. We empirically selected the lowest ranks for each dataset that provide good performance. We reported results with rank=25 for Small NORB, 30 for RaFD and 15 for 3dShapes. We have initialized the cores and bases for different tensor factorization using samples drawn from $N(0, 0.1)$ distribution. We kept fixed seed for every setup in order to achieve fair comparison. As we consider a batch of images correspond to a slice of a tensor core, we could perform minibatch optimization which reduced memory requirement during training. We have used Adam [29] optimization for network parameters optimization and Stochastic Gradient Descent (SGD) for optimizing tensor factorization parameters. The learning rate for Adam was selected to be 0.001 and SGD to be 1 (or 0.1). We let the optimization run for enough iterations to converge. For autoencoder setup, we set weight terms, λ_1 and λ_2 to be 1.

Table 1. Reconstruction quality (PSNR in dB) for image denoising and inpainting with different comparing approaches.

	LSTR	Deep Decoder	CSGM	CSAE	Tensor Ring AE
Denoising					
Small NORB	23.11	27.40	28.40	27.56	31.71
RaFD	20.30	25.14	26.67	29.66	32.10
3dShapes	20.13	28.06	28.63	33.52	35.97
Inpainting					
Small NORB	24.70	35.10	28.00	33.49	35.29
RaFD	21.31	31.87	24.83	31.91	33.55
3dShapes	21.92	35.22	27.26	36.8	39.43

Comparison: We solve the inverse/compressive sensing problems (e.g. denoising and inpainting) without ground truth images. We show comparison with 4 different baselines which also perform the same task.

CSAE: We use the encoder of a deterministic autoencoder to learn the latent space from the corrupted measurements and pass the learned latent codes through decoder and measurement matrices to match with the observed measurements. One can refer to the second term of Eqn (4) as the objective this approach minimizes. Eventually we learn the original image given the corrupted measurements without having the ground truth. We term it **CSAE** (Compressive Sensing with Auto Encoder).

CSGM: We tried to solve the compressive sensing problems given that we have a trained generative model with the learned distribution of the target data. It is similar to the work of Bora et. al. [21]. We term it **CSGM** following [21].

LSTR: We utilize the attribute information of the structured dataset and use it as a prior to minimize the least square measurement loss with SGD. We term it **LSTR** (Least Square minimization with Tensor Ring).

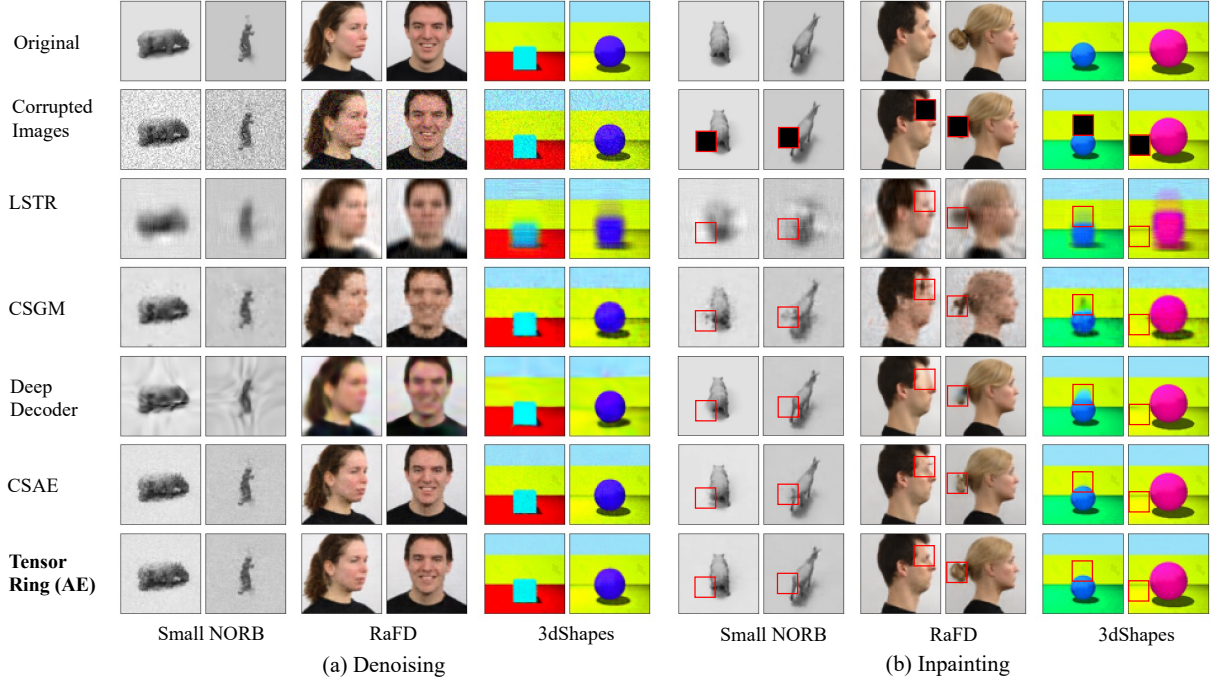


Fig. 2. Reconstruction results for (a) denoising and (b) inpainting on Small NORB, RaFD and 3dShapes datasets.

Deep Decoder: Finally we use one of the self supervised generative prior based approaches to solve the compressive sensing problems. We use **Deep Decoder** [22] for comparison.

We empirically demonstrate that our proposed Tensor Ring factorized Autoencoder outperforms all the four baselines in terms of reconstruction quality since we are using the advantages of both the structural information and generative priors.

Denoising: In this experiment, we added Gaussian noise of 20dB to all the images. We report the average reconstruction quality (dB PSNR) for different comparing techniques in Table 1. We also demonstrate some reconstructed images in Figure 2. We can observe that utilizing the structure in latent space helps us outperform the other approaches. Deep decoder uses a single network per image recovery. So it cannot use information from the other measurements of the structure. Although CSGM uses all the training data to train the generator, it does not explicitly use the structural information. We also observe that LSTR does not provide good reconstruction performance even though it is also using the structural information because images usually do not have the tensor structure in their representation. CSAE approach performs well as it learns the optimal embedding space for solving the inverse problem using an encoder. However, it falls behind our proposed approach since it does not use any structural information. By learning an embedding space to apply tensor structure, we are utilizing the structural information to our advantage.

Image Inpainting: It is often observed in real scenario

that some of the images of the structured image set are corrupted instead of being completely unavailable. We perform a set of experiments on different datasets where we missed a 16×16 block from all the images at random locations. We feed the structured image set to the AE based tensor factorized scheme. Our tensor factorized autoencoder utilize the strength of the structured organization of the dataset to better reconstruct the images with missing blocks. We report the reconstruction results in Table 1. We also demonstrate some reconstructions in Figure 2. We can observe that we outperform the other approaches especially in recovering the original details of missing blocks as shown in Figure 2. Although Deep Decoder and CSAE perform very close to our approach, they fail to recover reliable details in the missing blocks. Since deep decoder uses a separate network for every image, its memory and parameter requirements are significantly higher than our method.

4. CONCLUSION

We proposed tensorized autoencoder as a prior for solving compressive sensing problems. For structured datasets, we utilize the structural similarities in the images by applying tensor ring factorization in the latent space learned by an encoder. We demonstrated that applying structural constraint such as tensor ring performs better on the learned latent space. We also observed that by utilizing the structural similarity of the dataset, tensorized autoencoder can outperform other self supervised generative prior and deep image prior based approaches for different compressive sensing applications.

5. REFERENCES

- [1] Tamara G Kolda and Brett W Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
- [2] D Bacciu and DP Mandic, "Tensor decompositions in deep learning," in *28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2020*. ESANN (i6doc.com), 2020, pp. 441–450.
- [3] Yuwang Ji, Qiang Wang, Xuan Li, and Jie Liu, "A survey on tensor techniques and applications in machine learning," *IEEE Access*, vol. 7, pp. 162950–162990, 2019.
- [4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative Adversarial Networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.
- [5] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.
- [6] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam, "Optimizing the latent space of generative networks," in *Proc. Int. Conf. Machine Learning*, 2018.
- [7] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popovic, "Face transfer with multilinear models," in *ACM SIGGRAPH 2006 Courses*, pp. 24–es. 2006.
- [8] Zhiwei Deng, Rajitha Navarathna, Peter Carr, Stephan Mandt, Yisong Yue, Iain Matthews, and Greg Mori, "Factorized variational autoencoders for modeling audience reactions to movies," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2577–2586.
- [9] Xiaoqin Zhang, Di Wang, Zhengyuan Zhou, and Yi Ma, "Robust low-rank tensor recovery with rectification and alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 238–255, 2019.
- [10] Wenqi Wang, Yifan Sun, Brian Eriksson, Wenlin Wang, and Vaneet Aggarwal, "Wide compression: Tensor ring nets," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9329–9338.
- [11] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Tensor decomposition for compressing recurrent neural network," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.
- [12] Xinyu Chen, Zhaocheng He, and Lijun Sun, "A bayesian tensor decomposition approach for spatiotemporal traffic data imputation," *Transportation research part C: emerging technologies*, vol. 98, pp. 73–84, 2019.
- [13] Sriram Krishnaswamy and Mrinal Kumar, "Tensor decomposition approach to data association for multitarget tracking," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 9, pp. 2007–2025, 2019.
- [14] Emmanuel J Candes, Yonina C Eldar, Deanna Needell, and Paige Randall, "Compressed sensing with coherent and redundant dictionaries," *Applied and Computational Harmonic Analysis*, vol. 31, no. 1, pp. 59–73, 2011.
- [15] David L Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [16] Emmanuel J Candes and Terence Tao, "Decoding by linear programming," *IEEE transactions on information theory*, vol. 51, no. 12, pp. 4203–4215, 2005.
- [17] Richard Baraniuk and Philippe Steeghs, "Compressive radar imaging," in *Radar Conference, 2007 IEEE*. IEEE, 2007, pp. 128–133.
- [18] Fei Yang, Hong Jiang, Zuowei Shen, Wei Deng, and Dimitris Metaxas, "Adaptive low rank and sparse decomposition of video using compressive sensing," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*. IEEE, 2013, pp. 1016–1020.
- [19] Chen Zhao, Siwei Ma, Jian Zhang, Ruiqin Xiong, and Wen Gao, "Video compressive sensing reconstruction via reweighted residual sparsity," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 6, pp. 1182–1195, 2017.
- [20] M Salman Asif, Felix Fernandes, and Justin Romberg, "Low-complexity video compression and compressive sensing," in *2013 Asilomar Conference on Signals, Systems and Computers*. IEEE, 2013, pp. 579–583.
- [21] A. Bora, A. Jalal, E. Price, and A. Dimakis, "Compressed sensing using generative models," *Proc. Int. Conf. Machine Learning*, 2017.
- [22] R. Heckel and P. Hand, "Deep decoder: Concise image representations from untrained non-convolutional networks," *Proc. Int. Conf. Learning Representations (ICLR)*, 2018.
- [23] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Deep image prior," in *Proc. IEEE Conf. Comp. Vision and Pattern Recog. (CVPR)*, 2018, pp. 9446–9454.
- [24] D. Van Veen, A. Jalal, E. Price, S. Vishwanath, and Alexandros G. Dimakis, "Compressed sensing with deep image prior and learned regularization," *arXiv preprint arXiv:1806.06438*, 2018.
- [25] Tatsuya Yokota and Andrzej Cichocki, "Tensor completion via functional smooth component deflation," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 2514–2518.
- [26] Yann LeCun, Fu Jie Huang, and Leon Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*. IEEE, 2004, vol. 2, pp. II–104.
- [27] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg, "Presentation and validation of the radboud faces database," *Cognition and emotion*, vol. 24, no. 8, pp. 1377–1388, 2010.
- [28] Chris Burgess and Hyunjik Kim, "3d shapes dataset," <https://github.com/deepmind/3dshapes-dataset/>, 2018.
- [29] Diederik P Kingma and Jimmy Ba, "ADAM: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.