Image Generation Using a Sparsity Model

Gregory Vaksman and Michael Elad
Computer Science Department, Technion – Israel Institute of Technology

***Abstract*—In this work we present a novel method for generating images using sparse representations. The proposed method learns a dictionary and estimates the probability distribution of the sparse vectors from a given set of images. The probability distribution of the sparse vectors is represented as a product of two functions: one describing the support, and the other for the coefficients, given the support. We model the distribution of the support using a Markov network with pairwise factors, and assume that the atom coefficients, given the support, are normally distributed. We generate new images by drawing sparse vectors from the estimated probability and multiplying them by the pre-learned dictionary. We demonstrate the proposed method on three different sets of images: (i) MNIST, (ii) set of aligned faces, and (iii) set of unaligned faces from the MegaFace database.**

I. Introduction

In recent years we see increased interest in methods that are able to generate images. Texture images can be synthesized using several models, such as Gaussian [1], sparsity [2], [3], neural network [4], and other [5]–[8]. Of these, neural networks have been reported to have the ability to go beyond texture and generate images with content: handwritten digits, faces and even more complicated images [9]–[14]. Indeed, only few methods, excluding neural networks, show such a generative ability [15], [16]. When it comes to the sparsity model, only texture generations are reported [2], [3].

In this work we propose a novel sparsity based method for generating images with content. Given a set of example images, our model learns a dictionary, finds sparse representations of all the images in this set, and estimates the probability distribution of the sparse vector. New images are generated by drawing new sparse vectors from the estimated distribution, multiplying the obtained vectors by the dictionary and clipping the pixels values to range $[0, 255]$. Clearly, the greatest challenge is to estimate the Probability Density Function of the sparse representations. We present here an efficient method for achieving this goal.

II. Overview of the Proposed Model

The sparsity model assumes that images can be represented as linear combinations of the small number of atoms from some well-chosen dictionary. Given a set of N images $\{\underline{x}_i\}$, their sparse representations $\{\underline{\gamma}_i\}$ over some dictionary D, and the dictionary itself can be found by solving:

$$\min_{D,\{\underline{\gamma}^i\}} \sum_{i=1} \|\underline{x}^i - D\underline{\gamma}^i\|_2^2 \quad s.t\ \|\underline{\gamma}^i\|_0 \leq k, \quad 1 \leq i \leq N, \ (1)$$

where k is the maximum allowed number of atoms. Sparsity model has been shown to be useful for solving a variety of image restoration problems. However image generation using this model is not straightforward, because combinations of k randomly chosen atoms do not create valid images, see examples in Figure 1. Therefore, for the image generation task, some model that captures the statistics of the γ vectors, i.e. $P(\underline{\gamma})$, is needed. We can write $P(\underline{\gamma}) = P(\underline{a})P(\underline{u}|\underline{a})$, where $P(\underline{a})$ is a distribution of the support, and $P(\underline{u}|\underline{a})$ is a distribution of the coefficients, given the support. The vector $\underline{a}$ holds locations of the k non-zeros of the $\underline{\gamma}$, and the vector $\underline{u}$ contains their corresponding coefficients, $\gamma_{a_j} = u_j$ for $1 \leq j \leq k$. We restrict the coefficients to be positive, $u_j > 0$, and refer to positive and negative atoms as to different atoms, thereby doubling the dictionary size.

We model the support distribution $P(\underline{a})$ using fully connected Markov network with factors over all single variables $\psi(a_i)$ and over all pairs of variables $\phi(a_i, a_j)$ [17]. This model represents the probability function $P(\underline{a})$ as a normalized product of all factors, $P(\underline{a}) = \frac{1}{c}\prod_i \psi(a_i)\prod_{i,j}\phi(a_i, a_j)$, reducing the number of $P(\underline{a})$ parameters from M^k to about M^2, where M is a dictionary size. We note that the factors $\psi(a_i)$ and $\phi(a_i, a_j)$ are used only for reducing dimensionality, and have no meaning of marginal distributions. Unfortunately, in our case, the number of parameters is still significantly greater than the number of images, $M^2 \gg N$. Therefore we use an additional approximation replacing the factors $\psi(a_i)$ and $\phi(a_i, a_j)$ with marginal probabilities: $\psi(a_i) \approx P(a_i)$, and $\phi(a_i, a_j) \approx \frac{P(a_i,a_j)}{P(a_i)P(a_j)}$. Our final expression for $P(\underline{a})$ is

$$P(\underline{a}) = \frac{1}{c}\prod_i P(a_i)\prod_{i,j}\frac{P(a_i, a_j)}{P(a_i)P(a_j)} . \qquad (2)$$

The marginal pairwise probability $P(a_i, a_j)$ can be easily learned from data, since in N supports we have $0.5k(k-1)N$ pairs of atoms, which is quite a large number.

We assume that given the support, atom coefficients are normally distributed, i.e. $P(\underline{u}|\underline{a}) = \mathcal{N}(\underline{\mu}_{\underline{a}}, \Sigma_{\underline{a}})$. For generating a new image $\tilde{\underline{x}}$, we draw a support $\tilde{\underline{a}}$ from the $P(\underline{a})$ in Equation (2) using Gibbs sampling. Then we find intersections of $\tilde{\underline{a}}$ and supports of the given images, $\{\underline{b}^i\} = \{\underline{a}^i\} \cap \tilde{\underline{a}}$. We refer to the coefficients of the intersections $\{\underline{\gamma}_{\underline{b}^i}\}$ as data with missing values, and learn from them the $\underline{\mu}_{\tilde{\underline{a}}}$ and $\Sigma_{\tilde{\underline{a}}}$ using EM (Expectation Maximization) algorithm.

We apply the proposed scheme on three different sets of images: (i) MNIST [18], (ii) set of 4400 aligned faces of size 64×64 pixels, and (iii) set of 200,000 unaligned faces, downscaled to size of 64×64, from the MegaFace database [19]. For the MNIST we learn dictionary with K-SVD [20], and for faces with OSDL [21]. Simulation results are shown in Figures 2, 3 and 4.

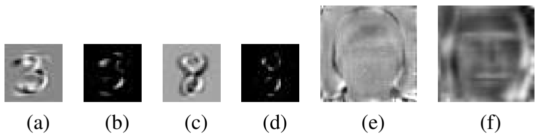

Fig. 1. Randomly chosen combinations of k atoms: 1(a) – digit *three* before clipping, 1(b) – digit *three* after clipping, 1(c) – digit *eight* before clipping, 1(d) – digit *eight* after clipping, 1(e) – aligned face, 1(f) – unaligned face. Note: both faces look very similar before and after clipping.

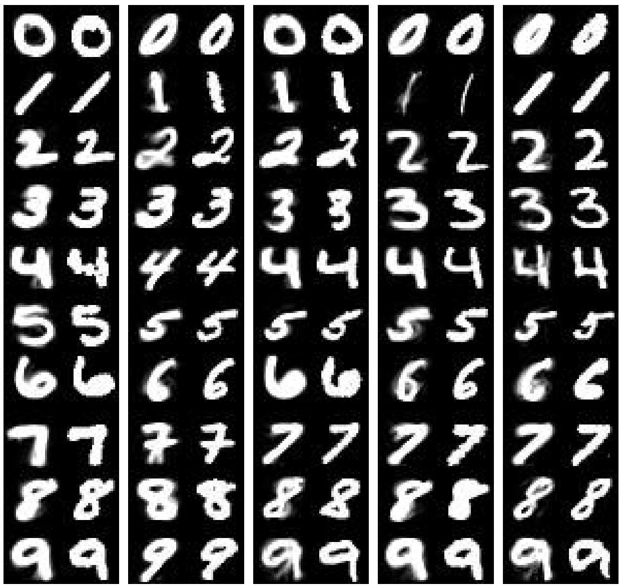

Fig. 2. In each column left digits are generated using our scheme, right digits are their nearest neighbors from the database.

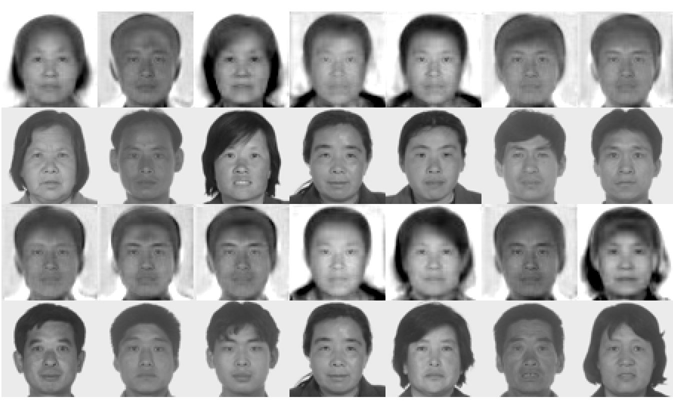

Fig. 3. Aligned faces. Faces in the first and third rows are generated using our method. The second and fourth rows show the nearest neighbors of the synthesized faces.

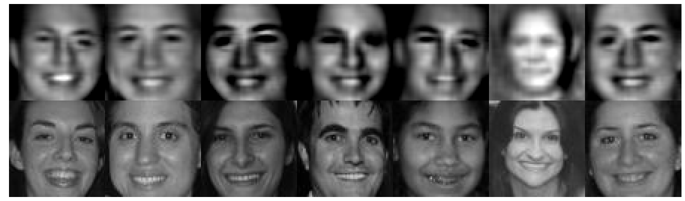

Fig. 4. Unaligned faces. Faces in the top row are generated using our method. The bottom row shows the nearest neighbors of the synthesized faces.

REFERENCES

[1] L. Raad, A. Desolneux, and J.-M. Morel, "Locally gaussian exemplar based texture synthesis," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 4667–4671.

[2] G. Peyré, "Sparse modeling of textures," *Journal of Mathematical Imaging and Vision*, vol. 34, no. 1, pp. 17–31, 2009.

[3] G. Tartavel, Y. Gousseau, and G. Peyré, "Variational texture synthesis with sparsity and spectrum constraints," *Journal of Mathematical Imaging and Vision*, vol. 52, no. 1, pp. 124–144, 2015.

[4] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 262–270.

[5] G. Peyré, "Texture synthesis with grouplets," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 32, no. 4, pp. 733–746, 2010.

[6] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 341–346.

[7] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," *ACM Transactions on Graphics (ToG)*, vol. 24, no. 3, pp. 795–802, 2005.

[8] S. Lefebvre and H. Hoppe, "Appearance-space texture synthesis," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 541–548, 2006.

[9] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[10] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *arXiv preprint arXiv:1601.06759*, 2016.

[11] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," *arXiv preprint arXiv:1502.04623*, 2015.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[13] T. Kim and Y. Bengio, "Deep directed generative models with energy-based probability estimation," *arXiv preprint arXiv:1606.03439*, 2016.

[14] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune, "Plug & play generative networks: Conditional iterative generation of images in latent space," *arXiv preprint arXiv:1612.00005*, 2016.

[15] C. Liu, H.-Y. Shum, and W. T. Freeman, "Face hallucination: Theory and practice," *International Journal of Computer Vision*, vol. 75, no. 1, pp. 115–134, 2007.

[16] Y. Ren, Y. Romano, and M. Elad, "Example-based image synthesis via randomized patch-matching," *arXiv preprint arXiv:1609.07370*, 2016.

[17] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT Press, 2009.

[18] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits," 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[19] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[20] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, Dec 2006.

[21] J. Sulam, B. Ophir, M. Zibulevsky, and M. Elad, "Trainlets: Dictionary learning in high dimensions," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3180–3193, 2016.