REGULARIZED GRADIENT DESCENT TRAINING OF STEERED MIXTURE OF EXPERTS FOR SPARSE IMAGE REPRESENTATION

Erik Bochinski, Rolf Jongebloed, Michael Tok, and Thomas Sikora

Technische Universität Berlin
Communication Systems Group
{*bochinski, jongebloed, tok, sikora*}@nue.tu-berlin.de

ABSTRACT

The Steered Mixture-of-Experts (SMoE) framework targets a sparse space-continuous representation for images, videos, and light fields enabling processing tasks such as approximation, denoising, and coding. The underlying stochastic processes are represented by a Gaussian Mixture Model, traditionally trained by the Expectation-Maximization (EM) algorithm. We instead propose to use the MSE of the regressed imagery for a Gradient Descent optimization as primary training objective. Further, we extend this approach with regularization terms to enforce desirable properties like the sparsity of the model or noise robustness of the training process. Experimental evaluations show that our approach outperforms the state-of-the-art consistently by 1.5 dB to 6.1 dB PSNR for image representation.

***Index Terms*—** Sparse Image Representation, Gaussian Mixture Model, Steered Mixtures of Experts, Denoising

1. INTRODUCTION

In recent years, machine learning based image and video representation techniques have received a lot of attention as they are able to learn structures and properties of such multimedia signals in a highly adaptive way. However, most of the research has been focused on artificial neural network based solutions [1–3], commonly trained using Gradient Descent. We explore how the same training techniques known from deep learning can be applied to the novel and well-defined *Steered Mixture-of-Experts* (SMoE) framework introduced in [4]. For such image coding applications SMoE yields compact sparse representations, allowing very efficient compression e.g. as only the parameters of a *Gaussian Mixture Model* (GMM) need to be stored. This unifying vision incorporates higher dimensional image modalities. In [5] and [6] the SMoE framework has been extended to video and light field representation and coding, respectively.

Unfortunately, learning models for higher qualities still poses a challenging task to be tackled. This is twofold reasoned. With increasing number of components the number of local minima increases. As such, the initialization of the parameters of the GMM (or SMoE in more general terms) becomes a crucial part [7]. GMMs are typically trained by the well-known *Expectation Maximization* (EM) algorithm which optimizes towards the maximum likelihood function [8] rather than minimizing the Mean Squared Error (MSE) of the regressed imagery. As a consequence, the maximum potential of such a model does not get fully exploited.

In [4], the joint likelihood of location and amplitude of pixels of the underlying image gets maximized. Deriving the conditional pdf such as in [9] called *Gaussian Mixture Regression* (GMR) the amplitude of a pixel given the location of that pixel can be regressed which leads to the alternative model of the *Mixture-of-Experts* (MoE) approach [10]. MoE approaches follow the divide-and-conquer principle. Each expert acts as a regression function weighted by a gating function. This arrives at soft partitioning of the input space to determine in which regions the experts are trustworthy. To avoid the problem of initialization of such a GMM in [7] a novel hierarchical Split-EM for yielding a sparse SMoE model has been presented which starts the modeling with few components.

This paper contributes to a framework which is able to optimize the underlying parameters towards minimizing the MSE using Gradient Descent (GD) [11]. In contrast to [7], the optimization starts with a very high number of components initialized on a evenly distributed grid. We introduce additional objectives to the loss function to establish a trade-off between the number of parameters and the regression er-

(a) Original (b) SMoE 34.63 dB (c) JPEG 34.68 dB

Fig. 1: Image quality comparison between regressed SMoE model trained by the proposed method and JPEG at the same PSNR level.

Accepted for publication at ICIP 2018

ror, promoting a sparsification of the model. Noise robustness is achieved by maximizing the bandwidth of the components.

The potential of SMoE can be seen in Fig. 1 which depicts reconstructed images of a crop of *Lena* at the same PSNR level using SMoE and JPEG, respectively. While JPEG (see Fig. 1c) suffers from several artifacts such as ringing and block artifacts, SMoE (see Fig. 1b) is visually more appealing. Detailed structures are preserved while noise corruption is reduced which induces the loss in PSNR compared to the original (see Fig. 1a).

2. STEERED MIXTURES OF EXPERTS

Gaussian Mixture Models are used to define a multivariate joint density distribution for (spatial input) random vector $\boldsymbol{x}$ and (luminance output) random variable y as sum of K weighted Gaussian distributions (kernels)

$$p\left(\boldsymbol{x}, y\right) = \sum_{i=1}^{K} \pi_i \cdot \mathcal{N}\left(\boldsymbol{x}, y; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) \tag{1}$$

with mixing coefficients π_i, for which $\sum\limits_{\forall i} \pi_i = 1$, covariance matrices and mean values (centers)

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} \boldsymbol{\Sigma}_{XX,i} & \boldsymbol{\Sigma}_{XY,i} \\ \boldsymbol{\Sigma}_{XY,i}^T & \sigma_{Y,i}^2 \end{bmatrix}, \boldsymbol{\mu}_i = \begin{bmatrix} \boldsymbol{\mu}_{X,i} \\ \mu_{Y,i} \end{bmatrix}. \tag{2}$$

When the model parameters are trained (i.e. by EM) a 2D regression function $y_p\left(\boldsymbol{x}\right)$ can be determined using the expected value of the conditional distribution of Y given $\boldsymbol{X}$

$$y_p\left(\boldsymbol{x}\right) = \mathbb{E}\left[Y|\boldsymbol{X}\right] = \sum_{i=1}^{K} m_i\left(\boldsymbol{x}\right) \cdot w_i\left(\boldsymbol{x}\right) \tag{3}$$

with the so called (hyper-) plane components or simply experts

$$m_i\left(\boldsymbol{x}\right) = \mu_{Y,i} + \boldsymbol{\Sigma}_{XY,i}^T \boldsymbol{\Sigma}_{XX,i}^{-1}\left(\boldsymbol{x} - \boldsymbol{\mu}_{X,i}\right) \tag{4}$$

and weighted soft max gating functions, called gates

$$w_i\left(\boldsymbol{x}\right) = \frac{\pi_i \cdot \mathcal{N}\left(\boldsymbol{x}, y; \boldsymbol{\Sigma}_{XX,i}, \boldsymbol{\mu}_{X,i}\right)}{\sum\limits_{j=1}^{K} \pi_j \cdot \mathcal{N}\left(\boldsymbol{x}, y; \boldsymbol{\Sigma}_{XX,j}, \boldsymbol{\mu}_{X,j}\right)}. \tag{5}$$

2.1. GMM Adaptations for Gradient Descent

A closer look on equation (4) and (5) reveals the independence of experts and gates. Although $\boldsymbol{\Sigma}_{XX,i}$ and $\boldsymbol{\mu}_{X,i}$ are part of both, the expert and gate definition, the necessary degrees of freedom to decouple each expert from its corresponding gate are introduced by $\mu_{Y,i}$ and $\boldsymbol{\Sigma}_{XY,i}^T$. Hence, each expert can be rewritten as

$$m_i\left(\boldsymbol{x}\right) = m_{0,i} + m_{1,i} \cdot x_{1,i} + m_{2,i} \cdot x_{2,i} \tag{6}$$

with $m_{0,i} = \mu_{Y,i} - \boldsymbol{\Sigma}_{XY,i}^T$, $[m_{1,i}\, m_{2,i}] = \boldsymbol{\Sigma}_{XY,i}^T \boldsymbol{\Sigma}_{XX,i}^{-1}$ and $[x_1\, x_2]^T = \boldsymbol{x}$. Thus, experts and gates can be trained independently if needed.

To enforce positive semidefiniteness of each covariance matrix $\boldsymbol{\Sigma}_{XX,i}$ and omit matrix inversion induced instabilities in the training process, we redefine each $\boldsymbol{\Sigma}_{XX,i}^{-1}$ by its Cholesky decomposition:

$$\boldsymbol{\Sigma}_{XX,i}^{-1} := \boldsymbol{A} \cdot \boldsymbol{A}^T \tag{7}$$

$$\boldsymbol{A} := \begin{pmatrix} a_{1,i} & 0 \\ a_{3,i} & a_{2,i} \end{pmatrix} \tag{8}$$

and optimize for $a_{1,i}$, $a_{2,i}$ and $a_{3,i}$ instead.

2.2. Multi-Task Optimization

The most commonly used training method for GMMs and MoEs in general is the EM algorithm. Unfortunately EM optimizes the likelihood of a mixture model to a given data set $(\boldsymbol{X}, Y)$. For regression purposes, it might be rather practical to optimize an underlying model by MSE directly. Thus, instead of taking the likelihood as quality metric, the MSE between image data y_n and parametric regression $y_p(\boldsymbol{x}_n)$ can form a more reasonable optimization criterion:

$$\mathcal{L}^{\text{MSE}} := \frac{1}{N} \sum_{n=1}^{N} \left(y_n - y_p\left(\boldsymbol{x}_n\right)\right)^2. \tag{9}$$

With an additional sparsity promoting regularization loss:

$$\mathcal{L}^{\text{S}} := \lambda_S \cdot \sum_{i=1}^{K} \pi_i \tag{10}$$

the mixing coefficients π_i are gradually decreased until values ≤ 0 are reached, stating that the respective kernel has no influence to the regression function and thus can be removed from the model. Analogously, the bandwidth of each kernel can be maximized by:

$$\mathcal{L}^{\text{D}} := \lambda_D \cdot \sum_{i=1}^{K} \frac{1}{|\Sigma_{XX,i}|} = \lambda_D \sum_{i=1}^{K} \left(a_{1,i} \cdot a_{2,i}\right)^2. \tag{11}$$

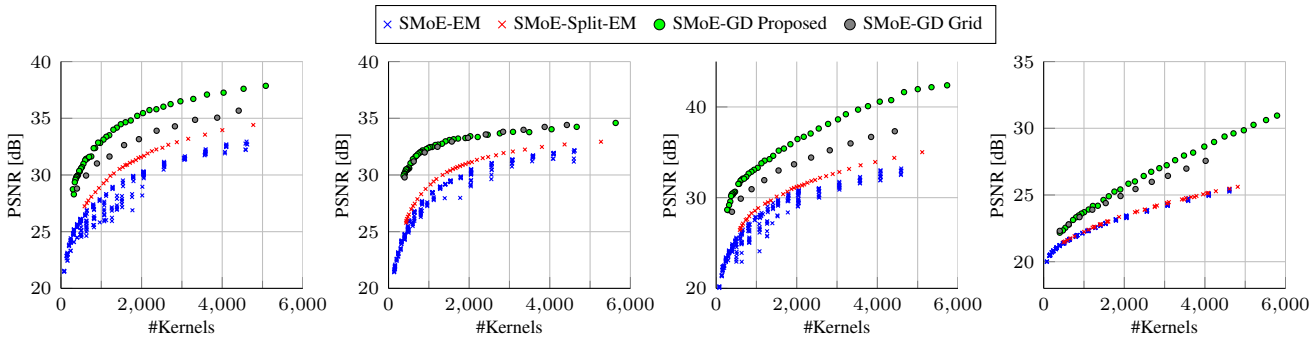
suppressing the modeling of too small details as usually introduced by noise. The final multi-task loss is composed as:

$$\mathcal{L} := \mathcal{L}^{\text{MSE}} + \mathcal{L}^{\text{S}} + \mathcal{L}^{\text{D}} \tag{12}$$

incorporating all above stated objectives. The influence of the sparsity and denosing losses $\mathcal{L}^{\text{S}}, \mathcal{L}^{\text{D}}$ can be adjusted by the respective coefficients λ_S and λ_D. The task then is to find a set of parameters $\boldsymbol{A}_i$, $\boldsymbol{m}_i$, $\boldsymbol{\mu}_i$ and π_i that minimizes $\mathcal{L}$:

$$\underset{\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{m}, \boldsymbol{\pi}}{\arg\min} \left\{\mathcal{L}\right\}. \tag{13}$$

Following the negative gradient $-\nabla\mathcal{L}$ via Gradient Descent is the most intuitive approach to such task.



(a) *Lena* (b) *Peppers* (c) *Cameraman* (d) *Baboon*

Fig. 2: Comparison of the image quality of regressed SMoE models learned using different training methods and initialization strategies for various numbers of used kernels in the models. SMoE-EM and SMoE-Split-EM refer to [4] and [7] respectively.

3. EXPERIMENTS

In this section we evaluate the performance of the proposed approach and compare the results to state-of-the-art methods. The implementation was done using the Tensorflow framework [12], the source-code is made publicly available[1]

In practice, the gradients of the variables to optimize are of different magnitudes, requiring different learning rates to leverage the potential of the SMoE framework. Training is performed using Adam [13] with learning rates of $10, 10^{-5}$ for $\boldsymbol{A}, \boldsymbol{\pi}$ respectively and 0.001 for $\boldsymbol{\mu}, \boldsymbol{m}$ if not stated otherwise. The remaining parameters are set to the default values of $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$ in all experiments.

3.1. Image Modeling

The process of modeling a training image is divided into three steps: First the model is initialized and pre-trained without any regularization. After the training is converged, the regularization to sparsify the model is employed. In the final step, the model is fine-tuned to minimize the regression MSE without regularization. Kernels are removed from the model when their respective mixing coefficients reach $\pi_i \leq 0$ during optimization in all stages.

Initialization & Pre-training The kernels of the model are initialized with $\boldsymbol{\mu}$ distributed evenly on a grid of $k \times k$ with $k = 128$ over the 512×512 pixels wide test images. All mixing coefficients are set to $\pi_i = \frac{1}{k^2}$. $\boldsymbol{A}$ is initialized in a way that the distance between the centers of two kernels equals two standard deviations 2σ. The experts are set to $m_{1,i} = m_{2,i} = 0$ and $m_{0,i}$ to the mean of the pixels of the training image where the gating of the respective kernel has the maximum influence. This results in $k^2 = 16384$ kernels which are trained for 10.000 iterations. No regularization is applied in this stage, so $\lambda_S = \lambda_D = 0$. However, some π_i reach values ≤ 0 removing the respective kernel from the model. Evaluation results after this stage are listed in the first line for each test image in Tab. 1.

[1]https://github.com/bochinski/tf-smoe

Regularization After pre-training, the regularization phase is employed. Best results are achieved by slowly introducing the regularization term and exponentially increasing the coefficient λ_S. Therefore, the following schedule is applied: $\lambda_S = \frac{x^2}{k^2}$ with x evenly distributed in [0.1,15] with 50 steps. After each step, the model is trained for an additional 1000 iterations. The fixed number of initial kernels k^2 is included to account for different sensitivity levels depending on the initialization of the kernels.

Fine-tuning In the last step, the model is trained for an additional 200 training iterations without any regularization to compensate for the prior trade-off between the MSE loss $\mathcal{L}^{\text{MSE}}$ and the regularization loss $\mathcal{L}^{\text{S}}$ to fine-tune the model towards a minimal MSE. Vanilla Gradient Descent with learning rates of $0.1, 10^{-8}$ for $\boldsymbol{A}, \boldsymbol{\pi}$ respectively and 10^{-6} for $\boldsymbol{\mu}, \boldsymbol{m}$ is used.

Results Fig. 2 and Tab. 1 show the achieved regression results for four common test images with different numbers of kernels in the model. Both reference methods are optimized using the EM algorithm instead of GD. SMoE-EM [4] initializes the model based on the richness of the textures in the image. SMoE-Split-EM [7] extends this approach by dynamically adding new kernels in areas which need higher attention. SMoE-GD Grid shows the results of our method without using any regularization and follows the described pre-training for various numbers of k. SMoE-GD Proposed refers to the above described setting. Apparently, GD optimization significantly outperforms EM based optimization in terms of PSNR and SSIM [14]. This may be reasoned by minimizing the regression MSE instead of maximizing the joint likelihood of location and amplitude of the pixels. Secondly, the proposed regularization scheme significantly outperforms the grid based optimization with the exception of the *Peppers* test image. This is most probably due to the fact that there are no areas that require special attention with more kernels, making an equal distribution of the kernels over the image sufficient. The fine-tuning stage consistently increases the regression quality of up to 0.7dB PSNR as shown in Tab. 1.

Table 1: Quality examples in terms of PSNR in dB and SSIM depending on the number of kernels per model. PSNR values for SMoE-GD Proposed are divided in results after regularization / fine-tuning phase.

Image	SMoE-GD Proposed			SMoE-GD Grid			GMM-Split-EM [7]		
	Kernel	PSNR	SSIM	Kernel	PSNR	SSIM	Kernel	PSNR	SSIM
Camera-	-	-	-	13378	45.91	0.99	-	-	-
man	3844	39.31 / **40.07**	0.96	3845	36.69	0.96	3947	33.92	0.91
	1928	35.68 / **36.02**	0.93	1933	33.67	0.93	1931	31.07	0.87
	880	32.72 / **32.81**	0.88	880	30.93	0.88	878	28.29	0.83
Lena	-	-	-	13056	39.85	0.96	-	-	-
	3876	36.92 / **37.17**	0.93	3876	35.04	0.92	4003	33.95	0.89
	1934	35.06 / **35.27**	0.9	1934	33.13	0.9	1921	31.05	0.85
	893	32.29 / **32.54**	0.87	893	31.02	0.86	854	28.48	0.79
Peppers	-	-	-	11899	36.36	0.91	-	-	-
	3868	33.86 / 33.96	0.84	3873	**34.24**	0.87	3805	32.47	0.81
	1971	33.14 / **33.41**	0.83	1971	33.27	0.84	1988	31.08	0.79
	896	31.84 / **31.94**	0.81	896	31.98	0.82	919	28.76	0.75
Baboon	-	-	-	11576	35.25	0.98	-	-	-
	3537	27.82 / **27.96**	0.8	3539	26.97	0.84	3532	24.76	0.63
	1907	25.34 / **25.46**	0.69	1907	24.92	0.72	1869	23.35	0.52
	889	23.42 / **23.46**	0.56	889	23.34	0.58	766	21.90	0.41

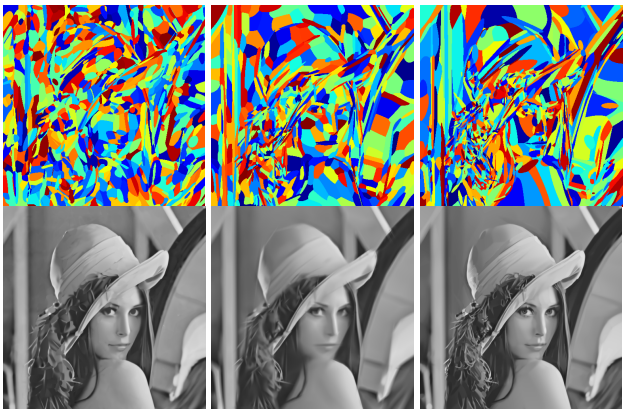
A visual comparison of the resulting models is shown in Fig. 3. For the grid initialization and no regularization, the spatial distribution of the kernel centers remain equal as seen in Fig. 3a. A concentration of kernels in the more highly textured area of the feather can be noticed for SMoE-Split-EM [7] in Fig. 3b. Our proposed SMoE-GD method (Fig. 3c) with regularization however shows the most desirable result by modeling the background of the image with only a few kernels while distributing the remaining kernels in the textured foreground.

3.2. Noise Robustness

Being robust against noise corruption is a crucial part in image modeling. Thus, we evaluate our SMoE training approach on a crop of *Peppers* corrupted by Additive White Gaussian Noise (AWGN) with a standard deviation of $\sigma = 10$ which is depicted in Fig. 4b and 4c showing the best results for $\lambda_D = 0$ and $\lambda_D = 2 \cdot 10^{-8}$ respectively, each with $k = 64$. In general, SMoE-GD is notably influenced by the noise corruption while the result in Fig. 4c benefits from the bandwidth regularization term. For comparison, results of the state-of-the-art denoising method BM3D [15] for AWGN with known σ are shown in Fig. 4d.

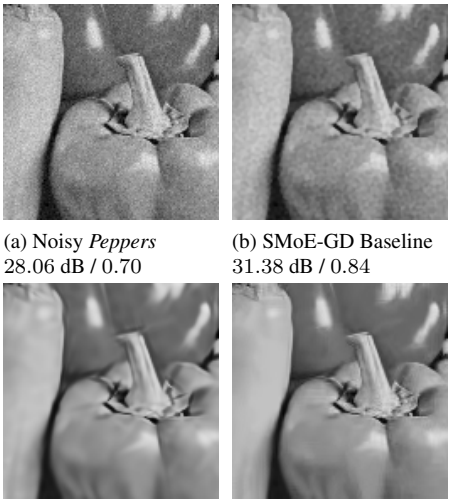
4. SUMMARY AND CONCLUSIONS

We proposed a novel, Gradient Descent based optimization strategy for the SMoE framework. This includes regularization to enforce desirable properties like the sparsity of the model or noise reduction characteristics of the regressed imagery. The experimental evaluation for the learning of sparse representation of images shows a significant improvement over the state-of-the-art methods with quality gains of up to 6.1 dB PSNR for the same number of model parameters. This is achieved by replacing the EM algorithm with Gradient Descent, optimizing the regression error directly. The regularization towards sparse models makes the method independent of sophisticated initialization techniques of previous approaches. Future research will include the exploration of how this promising performance gain can be leveraged for coding image data.



(a) SMoE-GD Grid $31.02dB/0.86$ (b) SMoE-Split-EM $28.67dB/0.80$ (c) SMoE-GD Proposed $32.48dB/0.87$

Fig. 3: Visualization and comparison of the kernel distribution with $K = 900$ for GD optimization with a grid initialization, the proposed method and EM optimization with splitting [7]. Each color in the top row codes the area of maximum influence of a respective kernel. The evaluation metrics are PSNR and SSIM. (Best viewed in color and zoomed-in)



(a) Noisy *Peppers* 28.06 dB / 0.70 (b) SMoE-GD Baseline 31.38 dB / 0.84

(c) SMoE-GD Proposed 32.26 dB / 0.90 (d) BM3D [15] 34.27 dB / 0.93

Fig. 4: Visualization of the SMoE noise reduction capabilities and comparison to the state-of-the-art (PSNR and SSIM). Figures (b) and (c) show the regressed SMoE model without and with bandwith regularization. The state-of-the-art denoising method BM3D [15] is included for reference (d).

5. ACKNOWLEDGEMENTS

This work was supported by a Google Faculty Research Award 2016 in Machine Perception.

6. REFERENCES

[1] Oren Rippel and Lubomir Bourdev, "Real-time adaptive image compression," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 2922–2930.

[2] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell, "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5306–5314.

[3] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszr, "Lossy image compression with compressive autoencoders," in *International Conference on Learning Representations*, 2017.

[4] Ruben Verhack, Thomas Sikora, Lieven Lange, Glenn Van Wallendael, and Peter Lambert, "A universal image coding approach using sparse steered mixture-of-experts regression," in *IEEE International Conference on Image Processing*, 2016, pp. 2142–2146.

[5] Lieven Lange, Ruben Verhack, and Thomas Sikora, "Video representation and coding using a sparse steered mixture-of-experts network," in *Picture Coding Symposium*, 2016.

[6] Ruben Verhack, Thomas Sikora, Lieven Lange, Rolf Jongebloed, Glenn Van Wallendael, and Peter Lambert, "Steered mixture-of-experts for light field coding, depth estimation, and processing," in *IEEE International Conference on Multimedia and Expo*, 2017, pp. 1183–1188.

[7] R. Jongebloed, R. Verhack, L. Lange, and T. Sikora, "Hierarchical Learning of Sparse Image Representations using Steered Mixture-of-Experts," in *2018 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2018.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[9] Hsi Guang Sung, *Gaussian mixture regression and classification*, Ph.D. thesis, Rice University, Houston, Texas, 2004.

[10] Seniha Esen Yuksel, J. N. Wilson, and P. D. Gader, "Twenty Years of Mixture of Experts," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1177–1193, 2012.

[11] Michael Tok, Rolf Jongebloed, Lieven Lange, Erik Bochinski, and Thomas Sikora, "An mse approach for training and coding steered mixtures of experts," in *Picture Coding Symposium (PCS)*, San Francisco, California USA, June 2018.

[12] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[13] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *International Conference for Learning Representations*, 2015.

[14] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[15] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.