Video Representation and Coding Using a Sparse Steered Mixture-of-Experts Network

Lieven Lange
Communication Systems Lab
Technische Universität Berlin
Berlin, Germany
lange@nue.tu-berlin.de

Ruben Verhack
Data Science Lab - iMinds
Ghent University
Ghent, Belgium
ruben.verhack@ugent.be

Thomas Sikora
Communication Systems Lab
Technische Universität Berlin
Berlin, Germany
sikora@nue.tu-berlin.de

***Abstract*—In this paper, we introduce a novel approach for video compression that explores spatial as well as temporal redundancies over sequences of many frames in a unified framework. Our approach supports "compressed domain vision" capabilities. To this end, we developed a sparse Steered Mixture-of-Experts (SMoE) regression network for coding video in the pixel domain. This approach drastically departs from the established DPCM/Transform coding philosophy. Each kernel in the Mixture-of-Experts network steers along the direction of highest correlation, both in spatial and temporal domain, with local and global support. Our coding and modeling philosophy is embedded in a Bayesian framework and shows strong resemblance to Mixture-of-Experts neural networks. Initial experiments show that at very low bit rates the SMoE approach can provide competitive performance to H.264.**

I. Introduction

Our challenge is the development of a bit-efficient video coder with easy bit-level access to MPEG-7-like low- and mid-level image features at the decoder for "compressed domain vision" applications [1][2][3]. Ideally, the compression strategy is designed such that also edge-preserving rate-conversion of the decoded video, such as spatio-temporal super-resolution enhancement or down-sampling, is intrinsically supported for the decoder. This calls for space-time-continuous and "edge-sensitive" sparse representations of image sequences - and description and redundancy reduction in the pixel domain rather than in the frequency domain.

In [2] we introduced the sparse *Steered Mixture-of-Experts* (SMoE) pixel-domain representation for still image compression. In this paper, SMoE is extended for coding video sequences. Our coding and modeling philosophy is deeply embedded in a Bayesian framework and involves training of Mixture-of-Experts networks. Representative parameters of the network are coded and used at the decoder to reconstruct the video sequences. The established DPCM/Transform algorithms currently used in all standard compression algorithms (i.e. MPEG, ITU [4]) explore temporal redundancies first. Motion vectors for individual blocks are estimated, coded and used for adaptive, linear prediction between frames. This is followed by spatial redundancy reduction via DCT or related linear transforms. Adaptivity to the underlying non-linear and/or in-stationary stochastic process is taken care of by adapting motion vectors in blocks and block and transform sizes within images and image sequences. A particular shortcoming of the DPCM/DCT approach is, however, that no sufficient long-term spatio-temporal correlation is explored this way. Especially at lower bit rates, the quantization artifacts introduced by coding the transform coefficients of the prediction errors causes annoying smearing and blocking artifacts in the decoded video sequences.

The SMoE network video compression approach outlined in this paper is motivated by the work on *Steering Kernel Regression* (SKR) [5] and our previous work on SKR for coding still images [6]. SKR produces excellent edge-preserving results for image denoising and super-resolution applications. For coding, however, the 'local' Gaussian kernels in SKR suffer from limited global support - the level of sparsity that can be achieved for coding is too limited. Our SMoE network kernels on the other hand are designed to provide local adaptability with global support. As such the SMoE network video coding approach introduced in this paper significantly departs from SKR in that the kernels that are employed are global steered (hyper-)volumes centered in irregular positions in the image domain. SMoE neural network approach has also a close relation to Support Vector Regression [7] and Radial Basis Functions Networks.

II. Steered Kernel Mixture-of-Experts Networks

Neural networks and kernel machines have been used as nonlinear adaptive systems for prediction, regression, classification and quality evaluation of video signals for many years [8][9][10]. Most recently Convolutional Neural Networks (CNNs) and Deep Learning strategies have boosted interest in this domain [11]. Non-linear and adaptive neural networks using Mixtures-of-Experts (MoE) are weighted combinations of K sub-networks [12][13]. In our work, we focus on this class of networks, because they can be designed such that easy interpretation and efficient coding of the model parameters becomes possible. In addition, hierarchical approaches are known so that Deep Learning strategies can be used to improve the approach in future work. We embed steering kernels in MoE sub-networks to make them highly adaptive to spatio-temporal variations in a video scene. In this way, the coded model parameters coincide with the desired MPEG-7-like features in images and video sequences. Previous work on

MoE for video in [14] concentrated on recurrent machines for segmentation. We keep the number of parameters significantly smaller than the number of measurement, which is important for a compact representation in a coding framework. Further, our strategy involves the optimization of the MoE by using all pixels in an entire sequence at once, to arrive at long-term motion representation. In this respect our coding approach follows the segmentation and classification framework in [15]. Work in [16] targets hierarchical segmentation for DPCM coding similar to H.264. We encode the kernel parameters directly and involve a regression framework at the decoder - no DPCM is performed.

We assume that the image sequence random process is modelled by a 4-D space-time-continuous stochastic model (3-D sample grid and pixel gray value amplitudes). The encoder modelling and analysis task involves training the parameters of the model. In general, regression attempts to optimally predict a realization of a random vector $Y \in \mathbb{R}^q$, based on a known random vector $X \in \mathbb{R}^p$. *Gaussian Mixture Models* (GMM) are frequently used to approximate multi-modal, multivariate distributions $p_{XY}(x, y)$. The parameters can be estimated from the training data by the *Expectation-Maximization* (EM) algorithm, which allows us to treat the learning process as a maximum likelihood (ML) problem [17][18][19]. Standard likelihood-based methodology can be used to train the networks in the first instance and to subsequently obtain confidence intervals for a predicted output corresponding to an input. Since we allow the Gaussian pdf functions to steer, we enable the desired steering regression capability. Each 4-D Gaussian component then acts as an "expert" in its respective arbitrarily-shaped spatio-temporal region of the image sequence. All K "experts" collaborate in a Mixture-of-Experts framework, thus one closed-form parametric, spatio-temporal, continuous regression function for the entire image sequence is derived.

This results in a Gaussian Mixture Regression (GMR) approach [20]. Assume that the image training data $D = \{x^i, y^i\}_{i=1}^N$ has the following joint probability density:

$$p_{XY}(X, Y) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mu_j, R_j) \quad \text{with} \sum_{j=1}^{K} \pi_j = 1 \tag{1}$$

The parameters of this model are $\mathbf{\Theta} = [\Theta_1, \Theta_2, ..., \Theta_K]$, with

$$\Theta_j = (\pi_j, \mu_j, R_j) \qquad \text{parameter set of } j\text{-th kernel}$$

$$\mu_j = \begin{bmatrix} \mu_{X_j} \\ \mu_{Y_j} \end{bmatrix} \qquad \text{mean of } j\text{-th kernel}$$

$$R_j = \begin{bmatrix} R_{X_jX_j} & R_{X_jY_j} \\ R_{Y_jX_j} & R_{Y_jY_j} \end{bmatrix} \qquad \text{covariance matrix of } j\text{-th kernel}$$

In our compression approach encoder, these parameters are quantized and coded (or transformed versions thereof, i.e. eigenvalues and eigenvectors). Many of those are strongly correlated between kernels. It is also possible to reduce the dimension of each kernel feature vector using standard PCA [21] or non-linear KPCA [22] approaches frequently employed in machine learning domain.

The decoder derives an analytical, $(p + q)$-dimensional regression function $m_j(x)$ for each expert using the decoded parameters and a confidence parameter vector σ_j^2

$$m_j(x) = \mu_{Y_j} + R_{Y_jX_j} R_{X_jX_j}^{-1} (x - \mu_{X_j}) \tag{2}$$

$$\sigma_j^2 = R_{Y_jY_j} - R_{Y_jX_j} R_{X_jX_j}^{-1} R_{X_jY_j} \tag{3}$$

Notice, that $m_j(x)$ is a linear hyper-volume in $\mathbb{R}^{p+q}$ with a $(p + q)$-dimensional slope defined by $R_{Y_jX_j} R_{X_jX_j}^{-1}$ – our desired linear steering kernel that provides global support over the entire video signal domain.

A signal at location x is estimated by the weighted sum over all K mixture components (4).

$$\hat{Y} = m(x) = \sum_{j=1}^{K} m_j(x) w_j(x) \tag{4}$$

and a weighting function

$$w_j(x) = \frac{\pi_j \mathcal{N}_j(\mu_{x_j}, R_{X_jX_j})}{\sum\limits_{i=1}^{K} \pi_i \mathcal{N}_i(\mu_{x_i}, R_{X_iX_i})} \tag{5}$$

Each expert defines the steered hyper-volume m_j, and a SoftMax "gating" window function w_j, which defines the operating region of the expert. The steered hyper-volume m_j describes a gradient, which indicates how the signal behaves around the center of the component (Equ. 2). The window function w_j gives weight to each sample, indicating the soft membership of that pixel to that component (Equ. 5). Every mode in the mixture model sub-network is treated as an expert and the experts collaborate towards the definition of the regression function. Note that the reconstruction is smoothed *piecewise linear*. By modelling the correlation between spatio-temporal sample location and amplitudes, our "local" SMoE components with "global" support can steer along spatio-temporal edges and adapt to regional signal intensity flow, similar to the "local" SKR [5]. In general, any $(p+q)$-dimensional regression can be preformed this way. Thus we could e.g. include color for video sequences into the regression formula, multiple views or the angular dimensions for lightfield images.

III. Coding and Embedded Functionality

The optimization problem for the EM algorithm used to estimate the parameters is unfortunately non-convex and converges to a local optimum [17]. Consequently, it is important to provide good initialization of the algorithm. We aim to arrive at few kernel components in regions that are flat, but a larger amount in detailed areas. Further, the total number of kernels generated impacts heavily on reconstructed image sequence quality as well as on the number of bits to code. We employ the sparsification approach presented in [2][6]. At the decoder side, for each kernel the coefficients of sub-matrix $R_{X_j X_j}$ and co-variance vector $R_{X_j Y_j}$ are needed for reconstruction of the image sequence.

Fig. 1 depicts the pixel reconstruction for a 32x32 image patch of test image ”Lena” with $K = 10$ EM optimized and coded SMoE 3-D kernels at 0.35 bpp [2]. It is apparent that the steered kernels allow excellent reconstruction of edges as well as the smooth transitions in the right part of the image patch even at such low rate. Also shown is the JPEG coded patch at same rate (without headers included in the counted bits). The advantage of a steering kernel network approach on image patches with strong edges is apparent. The Mixture model in (d) illustrates that the SMoE kernel network for a grey level image indeed consist of 3-D kernels that steer in horizontal, vertical as well as in grey level dimensions. In (e) the 2-D projection of the kernels shows the steering along the edges. Each kernel provides a steered plane for interpolation of the image in 3-D. The SoftMax windows derived using the SMoE approach are shown in (f). The windows are of arbitrary shape and provide steering capability. They can provide for sharp transitions between kernels (i.e. along edges) as well as for soft transitions in smooth areas (i.e. in the right half of the patch). Since our approach admits a Bayesian interpretation, a low-level segmentation of each image in a video sequence is readily derived from the SoftMax windows using ”hard decision” (g). The coded representation implicitly codes MPEG-7-like information about relevant data points and gray values (the centers of the kernel) as well as the 3-D gradients (h)(i).

The support of these “compressed domain vision” features readily extends into the temporal domain for SMoE coded video. Figure 2 (top) depicts the decoded 4-D SMoE kernels for 64 frames of video grey level test sequence MobilCalendar (128x128 pels crop) (see Fig. 5 for reference). The sequence contains predominantly translatory camera motion as well as diverse global moving objects (part of train and spinning mobile). Here the 4-D SMoE model (horizontal, vertical, temporal and grey level dimension) is reduced to visualize the 3-D sample grid representation using the marginal 3-D density. The coded centers that relate to pixel amplitudes are not depicted. It is apparent from the SMoE model that the kernels act like ”activation atoms” in 3-D space. Their location is adapted to the statistics of the signal and they steer spatially and in direction of the motion of the segments - thus providing a sophisticated motion clue. The temporal steering parameters are easily interpreted as the direction of short- or long-term motion within each kernel segment of arbitrary 3-D shape. In our representation, motion clues have global support over the entire 3-D image stack but are windowed in their impact by the (now 3-D) domain-overlapping SoftMax window functions. While many kernels provide support over the entire sequence, other kernels support new content appearing to the scene and occlusions. Notice, that no distinction is made between direction of temporal “motion” and spatial correlation - those concepts are one and the same in a unified spatio-temporal “intensity flow” framework. Using the multi-class Bayesian classification approach in Fig. 1 it is possible to derive a 3-D segment for each kernel domain that steers along direction of motion within the scene in form of a “tube”. As depicted in Fig. 2, some of those kernels obviously provide soft-windowed support over all frames of the video sequence. This is the case where image content in the first frame can still be seen in the last frame, even though motion shifted. Other kernel segments have only limited support both in spatial as well as in temporal direction. This indicates, that the ”linear” correlation model in temporal direction may not be adequate to model such non-linear motion trajectories in the scene. Such motion, however, can be easily tracked through a video scene by connecting temporally adjacent tubes with similar features (e.g. similar gray values). If such adjacent segments can not be found, this may indicate that new content appears into the scene (Intra content). Fig. 2 (bottom) depicts the steered kernels for 64 frames of a video sequence with identical frames (1st frame of the sequence in Fig. 2 (top). A small ratio of i.i.d. white Gaussian noise was added to each frame. EM optimization results in 500 kernels all located in the middle of the sequence and steered in same direction orthogonal to each frame. As with MPEG-like coding the SMoE kernel network coding approach results in coding one still image (because the rate for coding the temporal parameters is essentially zero).

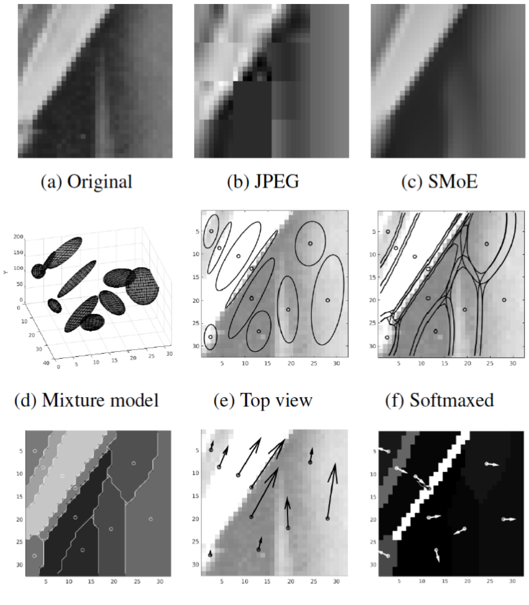
(a) Original
(b) JPEG
(c) SMoE
(d) Mixture model
(e) Top view
(f) Softmaxed
(g) Segmentation
(h) Intensity flow
(i) Edges

Fig. 1: Image patch coded with JPEG and SMoE with 10 kernels at same rate. Also shown are the steered kernels and the decoded MPEG-7-like feature descriptors for visual decoder processing.

Fig. 3 depicts selected frames of the colour MobileCalendar sequence and the EM modelling result with $K = 400000$ SMoE kernels each steered in 6-dimensions (3-D location 3-D color space for each kernel). It is apparent that the SMoE kernel network can reconstruct the sequence with high quality. The SMoE approach optimizes the kernel network by exploring correlation in the sequence with all 128x128x3x64

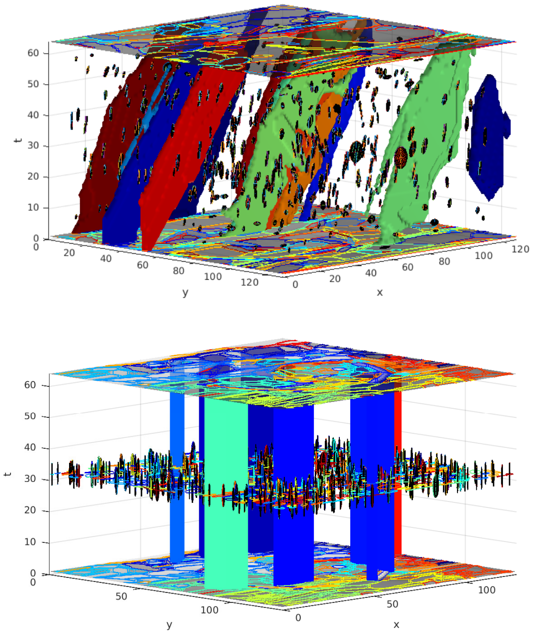

Fig. 2: Decoded SMoE model with $K = 500$ kernels for 64 frames of MobileCalendar (top). Same image in all frames (bottom).

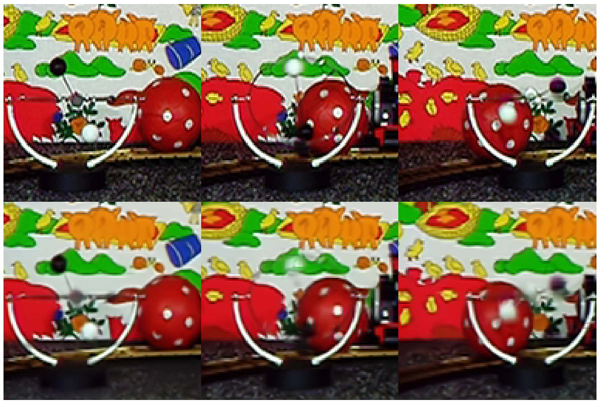

Fig. 3: Original (top) and SMoE (bottom), K=40000.

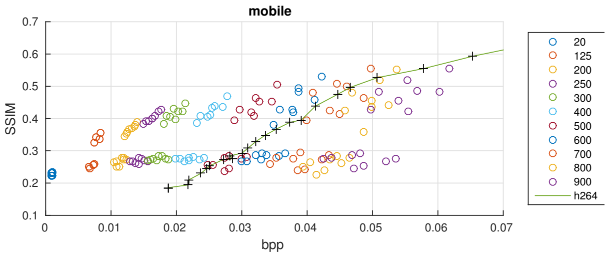

Fig. 4: R-D-Curves for sequence MobileCalendar.

TABLE I: Bit Rates and Quality Measures

	SMoE			H.264		
	Rate	PSNR	SSIM	Rate	PSNR	SSIM
MobileCal.	0.030	18.1	0.47	0.034	17.0	0.35
Race Horse	0.032	15.0	0.30	0.034	13.6	0.22
Race	0.030	20.1	0.66	0.032	18.2	0.57

pixel values processed in each iteration of the EM algorithm.

IV. Results for Video Coding

For coding we used crops of grey level test sequences MobileCalendar, RaceHorse and Race with 128x128 pixels, 64 frames and 25 Hz frame rate. RaceHorse and Race contain extremely fast camera and object motions. $K = 500$ kernels were estimated for each sequence and the EM algorithm was initialized randomly for the spatial positions and forced on the middle frame in temporal direction. The trained model (as in Fig. 2 for MobileCalendar) was used to reconstruct the SMoE video sequences.

Fig. 4 provides insight into the reconstruction quality (SSIM) vs various number of kernels K and quantization settings for sequence MobileCalendar. Results for other sequences were similar. At 0.055 bpp (appr. 22.5 kbit/s at 25 Hz frame rate), SSIM measures similar performance of SMoE and H.264. For higher rates (see also [2]) the SMoE approach cannot reconstruct fine details with reasonable bit efficiency. Essentially in its current implementation SMoE extends the capability of H.264 towards lower rates. SMoE approach outperforms H.264 for bitrates under 0.05 bpp and is even capable of going down to 0.008 bpp with recognizable video content.

Fig. 5 depicts results for each 10^{th} frame of sequences MobileCalendar and Horse at appr. 12 kbit/s (0.03 bpp). Table I shows the bit rates used as well as the average PSNR and SSIM results. At these low and ultra low rates H.264 fails to reconstruct any meaningful content at all, while SMoE still results in well recognisable content. This is also reflected in a drastic increase of PSNR and SSIM for SMoE. Due to the global kernel region-overlapping window approach no block artifacts are visible. The subjective quality gain of SMoE appears even more drastic when viewing the decoded video sequence in real-time playback. Because SMoE enables long-term motion representation, motion jerkiness and ”smearing” is completely avoided and salient objects are temporally smooth and clearly depicted. H.264 shows the typical DPCM smearing artifacts with new content coming into the scene and blocking artifacts. There are no motion clues that allow semantic understanding of the scene. The sequences can be found on our website (http://www.nue.tu-berlin.de/research/smoefvc).

V. Conclusions and Future Work

It is our challenge to provide a path towards novel coding strategies that may have the potential to outperform standard MPEG-like DCT/DPCM approaches. The presented SMoE networks offer such a novel strategy rooted in state-of-the-art machine learning algorithms. Our first experiments confirm that such networks may provide far better visual quality compared to DPCM/Transform approaches - presently at very low and ultra low rates. Additional MPEG-7-like features are embedded in the SMoE bitstream for further use. We

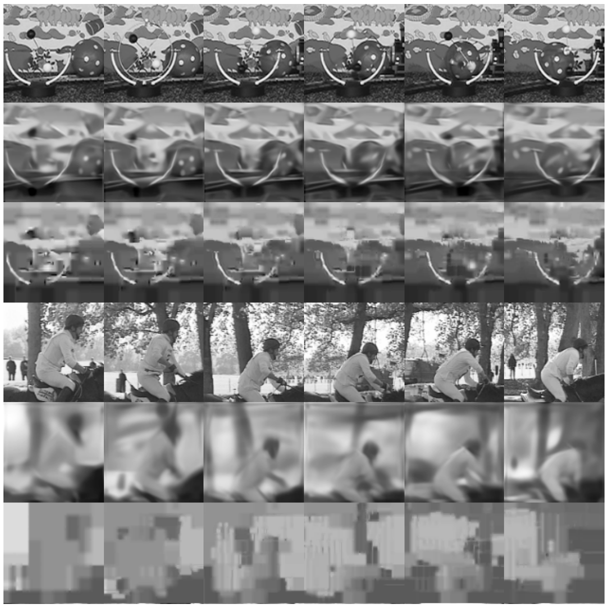

Fig. 5: Original, SMoE and H.264

emphasize that the comparison between SMoE and H.264 in this paper is neither conclusive nor can it be fair. H.261 is a highly optimized coder integrating highly adaptive processing strategies developed over a period of 30 years - only short-term motion redundancy is explored. SMoE is a network that explores long runs of successive frames for short- and long-term motion redundancy in a unified optimization approach, adapting automatically to local spatio-temporal statistics. Only basic and non-optimized coding strategies were used to compress the kernel parameters. To out knowledge the presented framework is the first one published that is optimized by embedding up to 64 frames in a video sequence in one batch process to arrive at long term redundancy reduction.

SMoE is thus far a very "shallow" network with just one hidden layer and cannot reconstruct fine texture details with reasonable bit efficiency. An extension towards a deeper network with capabilities to model textures for each of the K steering components is obvious next future work. This will alllow competition with H.264 at higher quality levels. Texture coding elements may be coded using 3-D Shape Adaptive DCT [23] or related strategies to provide additional features for each layer. No explicit shape coding needs to be embedded since SMoE "implicitly" encodes shape information with each kernel. Further bit rate savings are possible by improving the training/modeling part. The approach is "generic" in that any $(p+q)$-dimensional regression can be preformed using the SMoE approach. It is straightforward to apply SMoE for multiple views or the angular dimensions of light field images. Since the decoder arrives at a sparse continuous, parametric regression equation, a super-resolution or down-sampled version of video at any scale can be readily available. E.g. the spatio-temporal tubes in Fig. 2 can be reconstructed at any frame rate for fast forward or slow motion reconstruction without the need for motion-compensated rate conversion.

References

[1] T. Sikora, "The MPEG-7 visual standard for content description-an overview," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 696–702, jun 2001.

[2] R. Verhack, L. Lange, T. Sikora, G. Van Wallendael, and P Lambert, "A Universal Image Coding Approach Using Sparse Steering Regression with a Mixture of Experts," in *Su bmitted to ICIP 2016*, 2016.

[3] V. Mezaris, I. Kompatsiaris, N. Boulgouris, and M. Strintzis, "Real-time compressed-domain spatiotemporal segmentation and ontologies for video indexing and retrieval," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 5, pp. 606–621, 2004.

[4] T. Sikora, "Trends and Perspectives in Image and Video Coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 6–17, jan 2005.

[5] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *Image Processing, IEEE Transactions on*, vol. 16, no. 2, pp. 349–366, 2007.

[6] R. Verhack, A. Krutz, P. Lambert, R. Van de Walle, and T. Sikora, "Lossy image coding in the pixel domain using a sparse steering kernel synthesis approach," in *2014 IEEE International Conference on Image Processing (ICIP)*. oct 2014, pp. 4807–4811, IEEE.

[7] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[8] H. Cheng and D. Butler, "Segmentation of aerial surveillance video using a mixture of experts," in *Digital Image Computing: Techniques and Applications, 2005. DICTA '05. Proceedings 2005*, Dec 2005, pp. 66–66.

[9] H. Zhang, J. Yang, Y. Zhang, and T. S. Huang, "Non-local kernel regression for image and video restoration," in *Computer Vision–ECCV 2010*, pp. 566–579. Springer, 2010.

[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[11] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[12] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, "Adaptive Mixtures of Local Experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, feb 1991.

[13] S. Ng and G. McLachlan, "Using the EM algorithm to train neural networks: misconceptions and a new algorithm for multiclass classification," *Neural Networks, IEEE Transactions on*, vol. 15, no. 3, pp. 738–749, 2004.

[14] Y. Weiss and E.H. Adelson, "Motion estimation and segmentation using a recurrent mixture of experts architecture," in *Neural Networks for Signal Processing [1995] V. Proceedings of the 1995 IEEE Workshop*, Aug 1995, pp. 293–302.

[15] J. Goldberger H. Greenspan and A. Mayer, "Probabilistic space-time video modeling via piecewise gmm," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 3, pp. 384–396, 2004.

[16] G. Yazbek, C. Mokbel, and G. Chollet, "Video segmentation and compression using hierarchies of gaussian mixture models," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, April 2007, vol. 1, pp. I–1009–I–1012.

[17] T. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996.

[18] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[19] M. Jordan and L. Xu, "Convergence results for the EM approach to mixtures of experts architectures," *Neural Networks*, vol. 8, no. 9, pp. 1409–1431, jan 1995.

[20] H. Sung, *Gaussian Mixture Regression and Classification*, Ph.D. thesis, Rice University, 2004.

[21] I. Jolliffe, *Principal component analysis*, Springer series in statistics. Springer-Verlag, 1986.

[22] B. Schölkopf, A. Smola, and K. Müller, "Kernel principal component analysis," in *Artificial Neural Networks ICANN'97*, pp. 583–588. Springer, 1997.

[23] Thomas Sikora and Bela Makai, "Shape-adaptive dct for generic coding of video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 1, pp. 59–62, 1995.