

ELECTROENCEPHALOGRAM SENSOR DATA COMPRESSION USING AN ASYMMETRICAL SPARSE AUTOENCODER WITH A DISCRETE COSINE TRANSFORM LAYER

Xin Zhu* Hongyi Pan[†] Shuaiang Rong* Ahmet Enis Cetin*

*Department of Electrical and Computer Engineering, University of Illinois Chicago, USA

[†]Machine & Hybrid Intelligence Lab, Northwestern University, USA

ABSTRACT

Electroencephalogram (EEG) data compression is necessary for wireless recording applications to reduce the amount of data that needs to be transmitted. In this paper, an asymmetrical sparse autoencoder with a discrete cosine transform (DCT) layer is proposed to compress EEG signals. The encoder module of the autoencoder has a combination of a fully connected linear layer and the DCT layer to reduce redundant data using hard-thresholding nonlinearity. Furthermore, the DCT layer includes trainable hard-thresholding parameters and scaling layers to give emphasis or de-emphasis on individual DCT coefficients. Finally, the one-by-one convolutional layer generates the latent space. The sparsity penalty-based cost function is employed to keep the feature map as sparse as possible in the latent space. The latent space data is transmitted to the receiver. The decoder module of the autoencoder is designed using the inverse DCT and two fully connected linear layers to improve the accuracy of data reconstruction. In comparison to other state-of-the-art methods, the proposed method significantly improves the average quality score in various data compression experiments.

Index Terms— EEG signal sensor data compression, asymmetrical sparse autoencoder, discrete cosine transform, transform domain layer

1. INTRODUCTION

Electroencephalogram (EEG) plays a crucial role in neurological diagnosis, including epileptic illness diagnosis, brain inflammation, and dementia [1]. In existing literature, advanced feature extraction and classification algorithms have been designed for EEG analysis. Most of these algorithms gain insights into disease diagnosis by leveraging the abundance of EEG data detected by the sensor, which requires compression algorithms with high efficiency to establish high-capacity storage, fast transmission, and real-time analysis [2].

The current methods for compressing EEG signals are primarily categorized into three groups: traditional signal transform methods, neural network-based methods, and transform-based learning techniques. In [3], the discrete wavelet transform is applied to compress EEG data. It employed an optimization strategy to calculate the optimal control parameters which minimize the distortion and keep power consumption under a determined threshold. The work in

[4] developed a lossy compression model by utilizing the characteristics of epileptic EEG signals based on adaptive arithmetic encoding and discrete wavelet transform. However, these transform methods can not achieve a high reconstruction accuracy.

Neural network-based methods provide an alternative data compression approach by training models to learn the underlying patterns in the EEG signals. In [5], a convolutional autoencoder (CAE) is proposed to compress EEG data by employing convolutional layers and max-pooling layers to reduce the redundant data. Additionally, to retain more important information during the compression process, a new convolutional autoencoder is designed using a dynamic time warping (DTW) approximation as a loss function [6].

Transform-based learning methods have attracted considerable attention in the field of EEG compression over the past few decades [2]. A near-lossless compression algorithm is developed based on discrete cosine transform (DCT) and multilayer perceptron (MLP) in [7]. The energy concentration properties of DCT are leveraged to effectively reduce redundant data, while the MLP is utilized to compress the main DCT coefficients. Additionally, this approach calculates the reconstruction error to improve the accuracy. However, it does not have a good generalization ability in the transfer learning experiment.

To overcome the limitations of low reconstruction accuracy in transform-based methods and low compression efficiency in neural network-based methods, this paper proposes an asymmetrical sparse autoencoder with a DCT layer for EEG sensor data compression. We introduced the DCT and Hadamard transform domain layers into neural networks in a number of applications [8, 9]. In this paper, the key idea is to perform elementwise multiplications in the transform domain as convolutions in the time domain and use the well-known soft and hard-thresholding units [10] as the key nonlinearity of the network instead of the ReLU. In EEG data compression trainable thresholding units not only remove the noise in the data but also improve the data compression efficiency. A single fully connected or convolutional layer cannot be trained together with a soft-thresholding or hard-thresholding nonlinearity. However, by adding the fixed DCT after a fully connected layer we can not only train the thresholds but also train the fully connected layer which adapts the data and improves the data compaction capability of DCT. Furthermore, the DCT layer has trainable scaling parameters and a one-by-one convolutional layer. Scaling parameters approximately perform filtering in the transform domain and the one-by-one convolution layer reduces the dimension and produces the latent space where the EEG data is compressed. The decoder module reconstructs the EEG

This work was supported by NSF IDEAL 2217023.

waveform from compressed data by using Inverse DCT (IDCT) and two fully connected linear layers because the EEG decoder systems have more computational power compared to the encoder module.

The proposed model achieves the best compression efficiency and reconstruction accuracy compared with other autoencoder-based models. Experimental results show the proposed model outperforms other models on two EEG datasets: the BCI2 [11] and the Bonn University [12] datasets in terms of quality score. Additionally, the proposed model has low computation cost at the encoder side, which is suitable for implementation at the edge in sensors.

2. PRELIMINARIES

Our autoencoder structure has a DCT layer as shown in Fig. 1. The DCT [13] is a real-valued transform, which utilizes cosine functions as its basis functions. In this work, type-III DCT is employed because of its convolution property. Given an input vector $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]$, its orthogonal type-III DCT $\mathbf{X} = [X_0, X_1, \dots, X_{N-1}]$ is defined as:

$$X_k = \sqrt{\frac{1}{N}}x_0 + \sqrt{\frac{2}{N}} \sum_{n=1}^{N-1} x_n \cos \left[\frac{\pi}{N} \left(k + \frac{1}{2} \right) n \right], \quad (1)$$

Because of its good energy concentration properties, it has been widely used in many prevalent compression algorithms, including JPEG and MPEG algorithms [14, 15], where the DCT coefficients are weighted according to their importance before quantization. The weights are experimentally determined in these old standards. In this article, we will determine them using the backpropagation algorithm and call them scaling parameters. In fact, applying weights to DCT parameters is somewhat similar to the performing convolution in the Fourier transform because elementwise multiplication in the Fourier transform corresponds to circular convolution in the time domain [16]. The DCT convolution theorem also states the elementwise multiplication in the DCT domain corresponding to symmetric convolution in the time domain [17]. Formally, it can be written as:

$$\mathbf{x} \otimes_s \mathbf{w} = \mathcal{D}^{-1}(\mathcal{D}(\mathbf{x} \circ \mathbf{f}) \circ \mathcal{D}(\mathbf{w} \circ \mathbf{f})) \circ \mathbf{g}, \quad (2)$$

where $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{w} \in \mathbb{R}^N$ are input vectors. \circ represents the element-wise multiplication. \otimes_s denotes the symmetric convolution [18]. $\mathcal{D}(\cdot)$ and $\mathcal{D}^{-1}(\cdot)$ stands for the orthogonal type-III DCT and IDCT, respectively. \mathbf{g} is a constant vector:

$$\mathbf{g}[n] = \begin{cases} 1/(2\sqrt{N}), & n = 0, \\ \sqrt{1/(2N)}, & n > 0, \end{cases} \quad (3)$$

where $0 \leq n \leq N-1$. $\mathbf{f}[n] = 1/\mathbf{g}[n]$.

We experimentally observed that we do not need to perform symmetric convolutions to take advantage of the elementwise multiplications in the transform domain. We were able to train both the DCT domain weights and thresholding parameters using the backpropagation algorithm by inserting a fully connected layer before the DCT. We also observed that hard-thresholding performs better than soft-thresholding in EEG data compression.

3. ASYMMETRICAL SPARSE AUTOENCODER WITH A DCT LAYER

In this section, we describe the main building blocks of the proposed asymmetrical sparse autoencoder with a DCT layer (ASAEDCT).

3.1. Structure of ASAEDCT

Classical autoencoders are designed in a symmetric architecture, with a similar complexity on both the encoder and decoder sides [19]. The encoder side needs to perform heavy computations to compress the signals. The computational load from the encoder prevents the application of the autoencoders in low-cost sensors. In ASAEDCT, the encoder side has low complexity, but the decoder side has more linear layers than the encoder side to enhance the data reconstruction ability because the decoder can be implemented in a powerful host computer.

Encoder Part of the ASAEDCT:

The original EEG signal is partitioned into short-time windows of length N . This block of data is processed by the fully connected linear layer. Let the output of the linear layer be $\mathbf{x} \in \mathbb{R}^N$. The DCT $\mathbf{X} = [X_0, X_1, \dots, X_N]^T$ is computed using Eq. (1).

We have multiple channels to process the DCT data. Each channel corresponds to a different convolutional filter and each channel has hard-thresholding operators to perform denoising in the DCT domain. The proposed method applies the back-propagation algorithm to train hard-thresholding parameters and the transform domain weights. The hard-thresholding operator is defined as:

$$\tilde{X}_{i,k} = \mathcal{S}_{T_i}(X_k) + T_{i,k} \cdot \text{sign}(\mathcal{S}_{T_i}(X_k)), \quad (4)$$

where

$$\mathcal{S}_{T_i}(X_k) = \text{sign}(X_k) \cdot (|X_k| - T_{i,k})_+ \quad (5)$$

is the soft-thresholding function [10] and $T_{i,k}$ is a trainable threshold parameter for $0 \leq i \leq C$, $0 \leq k \leq N$; the subscript C stands for the number of channels; $(\cdot)_+$ is the rectified linear unit (ReLU) function. Although the soft-thresholding function can also perform transform domain denoising, it may also reduce the energy of large DCT coefficients. In EEG compression we found out that hard-thresholding produces better results.

Unlike the hand-crafted quantization matrix used in the JPEG standards [14], our network has trainable scaling vectors to assign suitable weights for different DCT coefficients. The DCT coefficients $\tilde{X}_{i,k}$ are element-wise multiplied by the scaling parameters:

$$\hat{X}_{i,k} = \tilde{X}_{i,k} \cdot V_{i,k}, \quad (6)$$

Another interpretation of the scaling layer is related to the DCT convolution theorem, i.e., C scaling channels correspond to C distinct convolution kernels in the time domain, and their combination forms a filter bank to enhance the ability to extract features.

The one-by-one convolutional layer [20] is used to reduce the number of channels and forms the latent space where the EEG data is compressed. After the scaling layer, tanh is employed to introduce a nonlinearity between the scaling layer and the one-by-one convolution layer. The overall operation in the DCT layer can be summarized in the Algorithm 1.

We observed that $C = 3$ produces the best results in EEG data compression (to be discussed in Section 4). Another observation that we had is that we obtained better coding results when we changed the order of scaling the hard-thresholding nonlinearity in our network as shown in Fig.1.

Decoder Part of the ASAEDCT:

Encoder output data is in the DCT domain and it is quantized as discussed in Subsection 3.3. Once the decoder receives the quantized data $\hat{\mathbf{X}}$ in the DCT domain, the decoder first computes its IDCT $\hat{\mathbf{x}}$.

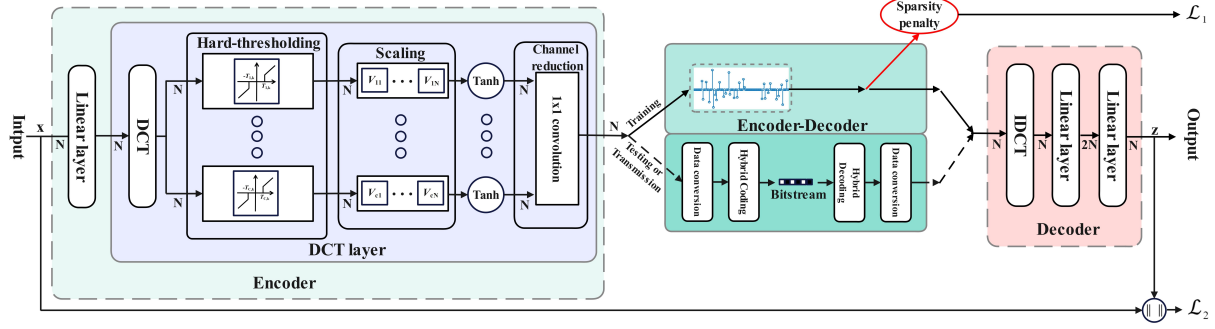


Fig. 1: Block diagram of the asymmetrical sparse autoencoder with a DCT layer.

Algorithm 1: C-path DCT layer

Input: Input tensor $\mathbf{x} \in \mathbb{R}^N$
Output: Output tensor $\mathbf{y} \in \mathbb{R}^N$
Define: $\mathbf{T}_i \in \mathbb{R}^N, \mathbf{V}_i \in \mathbb{R}^N, \mathbf{F} = \text{Conv1D}(\text{in} = C, \text{out} = 1, \text{kernel size} = 1)$;
 $\mathbf{X} = \text{DCT}(\mathbf{x}) \in \mathbb{R}^N$;
for $i = 1; i \leq C; i = i + 1$ **do**
 $\mathbf{U}_i = \text{sign}(\mathbf{X}_i) (|\mathbf{X}_i| - \mathbf{T}_i)_+ \in \mathbb{R}^N$;
 $\tilde{\mathbf{X}}_i = \mathbf{U}_i + \mathbf{T}_i \circ \text{sign}(\mathbf{U}_i) \in \mathbb{R}^N$;
 $\hat{\mathbf{X}}_i = \tilde{\mathbf{X}}_i \circ \mathbf{V}_i \in \mathbb{R}^N$;
 $\mathbf{R}_i = \tanh(\hat{\mathbf{X}}_i) \in \mathbb{R}^N$;
end
 $\mathbf{R} = \text{stack}(\mathbf{R}_i) \in \mathbb{R}^{C \times N}$;
 $\mathbf{y} = \mathbf{F}(\mathbf{R}) \in \mathbb{R}^N$;
return $\mathbf{y} \in \mathbb{R}^N$;

After this stage, the output of the network is obtained using two fully connected layers as shown in Fig. 1. Compared with the encoder, more linear fully connected layers are included in the decoder to enhance the reconstruction ability.

3.2. Sparsity Penalty Based Cost Function

In this subsection, we describe the cost function used in training the proposed ASAEDCT system. Sparsity constraints [21] are introduced to keep the feature map sparse, which enhances the compression efficiency of the DCT layer. Suppose the output of the one-by-one convolutional layer is $\mathbf{y} \in \mathbb{R}^N$, the activity of y_j is defined as:

$$\hat{y}_j = \sigma(\mathbf{y})_j = \frac{e^{|y_j|}}{\sum_{k=0}^{N-1} e^{|y_k|}}, i = 0, 1, \dots, N-1, \quad (7)$$

where $\sigma(\cdot)$ stands for the softmax function. Next, the Kullback-Leibler divergence (KLD) [21] is utilized as the sparsity penalty term because it can measure the difference between two probability distributions. The KLD is defined as

$$\sum_{j=0}^{N-1} \text{KL}(\alpha || \hat{y}_j) = \sum_{j=0}^{N-1} \alpha \log \frac{\alpha}{\hat{y}_j} + (1 - \alpha) \log \frac{1 - \alpha}{1 - \hat{y}_j}, \quad (8)$$

where α is a sparsity parameter. Therefore, the overall loss function \mathcal{L} is composed of the linear combination of the mean squared error and the KLD:

$$\mathcal{L} = \frac{1}{N} \sum_{i=0}^{N-1} (x_i - z_i)^2 + \lambda \sum_{j=0}^{N-1} \text{KL}(\alpha || \sigma(\mathbf{y})_j), \quad (9)$$

where λ is the weight of the sparsity penalty term. As shown in Fig 1, x_i and z_i represent the input and reconstructed signals, respectively. In the training phase, a threshold ξ is employed to adjust the compression ratio. When the proportion of 0's in \mathbf{y} is lower than ξ , the training is terminated. In this manner, small entries (redundant information) are eliminated by being set to 0. The setting of ξ is presented in Section 4.

3.3. Data Encoding and Storage

As is shown in Fig. 1, the data transmission module consists of data conversion and hybrid coding. The hybrid coding algorithm is developed using a combination of Run-Length Encoding (RLE) [22] and Lempel-Ziv-Markov chain algorithm (LZMA) [23]. Each double floating-point data requires a storage space of 64 bits in the memory, whereas an integer only needs 32 bits [24]. Therefore, floating-point numbers are converted to integers. The data conversion is formulated as:

$$\hat{\mathbf{y}} = \text{Round}(10^\theta \times \mathbf{y} / \phi), \quad (10)$$

where $\text{Round}(\cdot)$ is the integer rounding function; θ and ϕ are both integers. They are used to adjust the compression ratio. This data is encoded into a bitstream for wireless transmission or storage into an ambulatory device using the RLE and LZMA. The RLE is first employed to eliminate consecutive repeated zeros in the latent space. Then, LZMA is implemented on the output of the RLE. At the decoder, this bitstream is converted back into integers and processed by the decoder part of ASAEDCT for signal reconstruction in the host computer.

4. EXPERIMENTAL RESULTS

In this work, the BCI2 dataset [11], and the Bonn University dataset [12] are used to evaluate the performance of the proposed EEG compression scheme. (1) The BCI2 dataset [11] is collected from a normal subject using 28 EEG channels. The sampling frequency is 100Hz. It contains 316 training sets and 100 testing

sets. Each channel includes 50 samples. (2) The Bonn University dataset [12] includes five records, *i.e.*, F, N, O, Z, S. To evaluate the generalization ability of the model, we use record S which is collected from an epileptic subject. All data sets are partitioned into blocks with a size of $N = 64$. The AdamW optimizer [25] is utilized in the training process. We choose θ , ϕ and ξ as 4, 5 and 0.6 because it can achieve a balance between compression efficiency and reconstruction accuracy. Additionally, the batch size is 16, and the learning rate is 0.001. λ is 10. Moreover, we use the compression ratio (CR) [26] and percent root-mean-square difference (PRD) [27] as the performance metrics. The comparison efficiency ascends as the CR increases, and the difference between the constructed signal and the original signal reduces as the PRD decreases.

To validate the compression performance of the proposed model, we compare our model with the ANN+DCT [7]. Additionally, we consider the transform-based method [27], the neural networks-based method [21] and the transform-based learning method [28] for further comparison as they both have a low complexity encoder.

Table 1 provides the detailed data compression performance of the proposed methods versus the other four state-of-the-art algorithms on the BCI2 datasets. In comparison to sparse AE, the proposed approach increases CR from 21.38 to 25.66 and decreases PRD from 8.51 to 5.09. Additionally, our model is superior to DCT-DOST because it has a higher CR and a lower PRD. It indicates that ASAEDCT which combines the orthogonal DCT transform and a neural network can achieve a better compression performance. Compared with other transform-based learning compression methods [7, 28], ASAEDCT still provides a better quality score. This is because the DCT layer improves compression efficiency and asymmetrical structure enhances the reconstruction ability in the decoder. Moreover, to validate the robustness of the model, the model trained using the BCI2 dataset is also tested on the Bonn University dataset. It is observed that the AAESDCT with $C = 3$ channels outperforms other methods as it achieves the best QS. Therefore, the model has a good generalization capability in data compression tasks.

Test Data	Algorithm	CR	PRD (%)	QS
BCI2	AAN+DCT [7]	21.57	6.38	3.38
	Sparse AE [21]	21.38	8.51	2.51
	DCT-DOST [27]	21.67	10.40	2.08
	AE-DCT [28]	21.95	9.18	2.39
	ASAEDCT	25.66	5.09	5.05
Bonn	AAN+DCT [7]	4.45	7.22	0.62
	Sparse AE [21]	6.76	6.18	1.09
	DCT-DOST [27]	7.56	11.09	0.68
	AE-DCT [28]	6.84	6.65	1.03
	ASAEDCT	8.40	5.89	1.42

Table 1: Compression experimental results. All methods are trained on the BCI2 dataset.

For visual evaluation of the compression performance of the proposed methods, the original signals and reconstruction signals are depicted in Fig 2. It is observed that the original signals and reconstruction signals are similar to each other as the differences between them are limited to a small range.

Table 2 presents the ablation study of each module in ASAEDCT. When the scaling layer, sparsity penalty term, or nonlinearity are

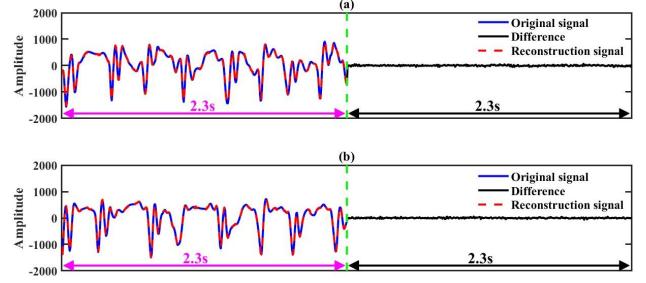


Fig. 2: Comparison of the time domain between the original data and the reconstructed data on the Bonn dataset.

Algorithm	CR	PRD (%)	QS
No nonlinearity	25.12	5.17	4.86
No scaling	25.46	11.79	2.16
No penalty	24.51	5.26	4.66
Only DCT	25.44	7.05	3.61
With soft-threshold	25.34	11.62	2.18
One-channel ($C=1$)	25.07	6.00	4.18
Two-channel ($C=2$)	24.90	5.39	4.62
ASAEDCT($C=3$)	25.66	5.09	5.05
Four-channel ($C=4$)	25.01	5.31	4.71

Table 2: Ablation experimental result.

removed, the CR decreases and PRD increases, leading to a lower QS. Additionally, the three-channel ($C = 3$) model is superior to one-channel, two-channel, and four-channel models. When we use only the DCT to compress the EEG data, the QS decreases. When hard thresholds are replaced with soft thresholds, the performance degrades.

Lastly, an analysis of the computational cost related to the proposed method is conducted. For an input with a duration of 6.4s, the compression time of the ASAEDCT network is only around 0.016s. Although this experiment is executed on the Intel Core i7-12700H CPU, the fast compression time indicates that the proposed method can be used in real-time compression of the EEG signals.

5. CONCLUSION

This study presents an asymmetrical autoencoder with a DCT layer that contains hard-thresholding nonlinearity for EEG sensor data compression. Since the encoder module uses the combination of one fully connected linear layer and the DCT layer the threshold values and DCT domain weights can be trained using a backpropagation type algorithm. The hard-thresholding nonlinearity and scaling layers not only enhance the data compaction capability of the DCT layer but also denoise the EEG signal. The decoder module based on the IDCT and two linear layers reconstructs the original signal. The proposed model achieves the best compression efficiency and reconstruction accuracy compared to other methods in BCI2 and Bonn EEG datasets. The computational load of the compression part of the ASAEDCT network is low, therefore, it can be implemented in the sensor for EEG data compression.

6. REFERENCES

- [1] Kahlaa K Al-Nassrawy, Ali Kadhum Idrees, and Dhiah Al-Shammary, "A novel lossless EEG compression model using fractal combined with fixed-length encoding technique," in *AI and IoT for Sustainable Development in Emerging Countries: Challenges and Opportunities*, pp. 439–454. Springer, 2022.
- [2] A Enis Cetin and Hayrettin Köymen, "Compression of digital biomedical signals," in *The Biomedical Engineering Handbook: Medical Devices and Systems*, pp. 3–1. CRC Press, 2006.
- [3] Ramy Hussein, Amr Mohamed, and Masoud Alghoniemy, "Scalable real-time energy-efficient EEG compression scheme for wireless body area sensor network," *Biomedical Signal Processing and Control*, vol. 19, pp. 122–129, 2015.
- [4] Binh Nguyen, Wanli Ma, and Dat Tran, "A study of combined lossy compression and seizure detection on epileptic EEG signals," *Procedia Computer Science*, vol. 126, pp. 156–165, 2018.
- [5] Abeer Z Al-Marridi, Amr Mohamed, and Aiman Erbad, "Convolutional autoencoder approach for EEG compression and reconstruction in m-health systems," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2018, pp. 370–375.
- [6] Hugo Lerogeron, Romain Picot-Clément, Laurent Heutte, and Alain Rakotomamonjy, "Learning an autoencoder to compress EEG signals via a neural network based approximation of DTW," *Procedia Computer Science*, vol. 222, pp. 448–457, 2023.
- [7] Behzad Hejrati, Abdolhossein Fathi, and Fardin Abdali-Mohammadi, "A new near-lossless EEG compression method using ANN-based reconstruction technique," *Computers in Biology and Medicine*, vol. 87, pp. 87–94, 2017.
- [8] Hongyi Pan, Daa Badawi, Chang Chen, Adam Watts, Erdem Koyuncu, and Ahmet Enis Cetin, "Deep neural network with Walsh-Hadamard transform layer for ember detection during a wildfire," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 257–266.
- [9] Daa Badawi, Agamyrat Agambayev, Sule Ozev, and A Enis Cetin, "Discrete cosine transform based causal convolutional neural network for drift compensation in chemical sensors," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 8012–8016.
- [10] David L Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.
- [11] Benjamin Blankertz, Gabriel Curio, and Klaus-Robert Müller, "Classifying single trial EEG: Towards brain computer interfacing," *Advances in Neural Information Processing Systems*, vol. 14, 2001.
- [12] Ralph G Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Physical Review E*, vol. 64, no. 6, pp. 061907, 2001.
- [13] Nasir Ahmed, T. Natarajan, and Kamisetty R Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.
- [14] Gregory K Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.
- [15] Praveen Sripada, "MP3 decoder in theory and practice," 2006.
- [16] Yuhao Xu and Hideki Nakayama, "DCT-based fast spectral convolution for deep convolutional neural networks," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.
- [17] Stephen A Martucci, "Symmetric convolution and the discrete sine and cosine transforms," *IEEE Transactions on Signal Processing*, vol. 42, no. 5, pp. 1038–1051, 1994.
- [18] HyunWook Park, YoungSeo Park, and Seung-Kyun Oh, "L/M-fold image resizing in block-DCT domain using symmetric convolution," *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1016–1034, 2003.
- [19] Kimia Dinashi, Ali Ameri, Mohammad Ali Akhaee, Kevin Englehart, and Erik Scheme, "Compression of EMG signals using deep convolutional autoencoders," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 7, pp. 2888–2897, 2022.
- [20] Min Lin, Qiang Chen, and Shuicheng Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.
- [21] Andrew Ng et al., "Sparse autoencoder," *CS294A Lecture Notes*, vol. 72, no. 2011, pp. 1–19, 2011.
- [22] Shahin Akhter and MA Haque, "ECG compression using run length encoding," in *2010 18th European Signal Processing Conference*. IEEE, 2010, pp. 1645–1649.
- [23] Zongjie Tu and Shiyong Zhang, "A novel implementation of JPEG 2000 lossless coding based on LZMA," in *The Sixth IEEE International Conference on Computer and Information Technology (CIT'06)*. IEEE, 2006, pp. 140–140.
- [24] Paruj Ratanaworabhan, Jian Ke, and Martin Burtscher, "Fast lossless compression of scientific floating-point data," in *Data Compression Conference (DCC'06)*. IEEE, 2006, pp. 133–142.
- [25] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
- [26] Hardev Singh Pal, Anil Kumar, Amit Vishwakarma, and Girish Kumar Singh, "Optimized tunable-q wavelet transform based 2d ECG compression technique using DCT," *IEEE Transactions on Instrumentation and Measurement*, 2023.
- [27] Chandan Kumar Jha and Maheshkumar H Kolekar, "Electrocardiogram data compression using DCT based discrete orthogonal Stockwell transform," *Biomedical Signal Processing and Control*, vol. 46, pp. 174–181, 2018.
- [28] Hongyi Pan, Xin Zhu, Zhilu Ye, Pai-Yen Chen, and Ahmet Enis Cetin, "Real-time wireless ECG-derived respiration rate estimation using an autoencoder with a DCT layer," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.