

Image-based Data Representations of Time Series: A Comparative Analysis in EEG Artifact Detection

Aaron Maiwald^{a,1}, Leon Ackermann^{a,1}, Maximilian Kalcher^a, Daniel J. Wu^{b,*}

^a*University of Osnabrück Department of Cognitive Science, Wachsbleiche
27, Osnabrück, 49090, Lower Saxony, Germany*

^b*Stanford University Department of Computer Science, 353 Jane Stanford
Way, Stanford, 94305, CA, USA*

Abstract

Alternative data representations are powerful tools that augment the performance of downstream models. However, there is an abundance of such representations within the machine learning toolbox, and the field lacks a comparative understanding of the suitability of each representation method.

In this paper, we propose artifact detection and classification within EEG data as a testbed for profiling image-based data representations of time series data. We then evaluate eleven popular deep learning architectures on each of six commonly-used representation methods.

We find that, while the choice of representation entails a choice within the tradeoff between bias and variance, certain representations are practically more effective in highlighting features which increase the signal-to-noise ratio of the data. We present our results on EEG data, and open-source our testing framework to enable future comparative analyses in this vein.

Keywords: deep learning, alternative data representations, time series representations, artifact detection, TUH EEG

*Corresponding Author.

Email address: danjwu@cs.stanford.edu (Daniel J. Wu)

¹These authors contributed equally to this work.

1. Introduction

Data transformations are essential tools for machine learning researchers and practitioners. The transformation of raw data into alternative representations often unlocks downstream learning. These transformations are not just academic exercises; they are essential tools, with proven effectiveness in enhancing interpretability, efficiency, and model performance across a wide variety of disciplines.

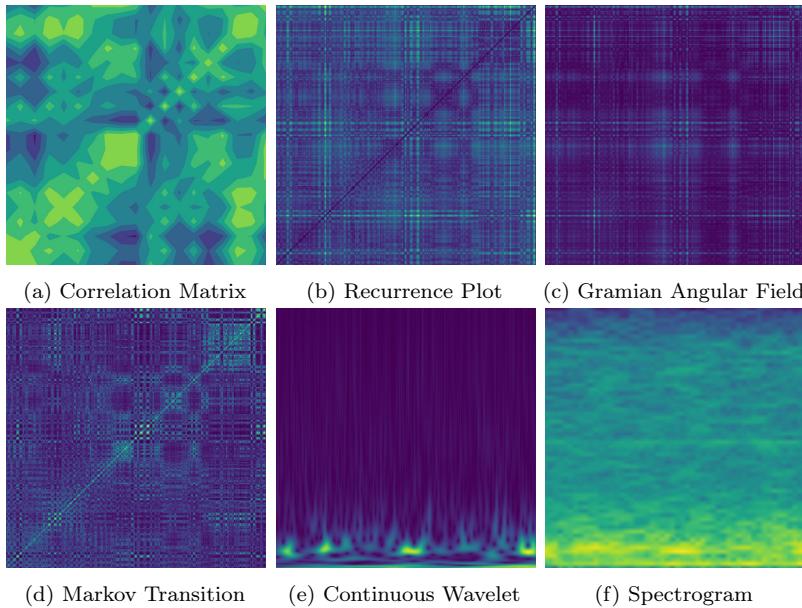


Figure 1: Exemplar image representations of time series EEG data.

There is a wide panoply of such transformations within the research scientists' toolbox, and a comparative study of such transformations is useful for enabling efficient and effective future research.

As such, our work makes three primary contributions:

- We introduce artifact detection in EEG data as a useful toy classification problem for assessing alternative data representation methods.
- We profile the characteristics and performance of six time series data representation methods on our toy problem, across a wide range of deep learning model architectures.

- We opensource our testing framework and testbed to facilitate future investigations into data representation methods.

In Section 2, we introduce alternative data representations in machine learning, and a toy problem to analyze time series data representations. Section 3 presents our dataset and six data representation methods to be examined, while Section 4 describes our experimental pipeline. Section 5 presents the results of our experiments, which are discussed in Section 6. Section 7 concludes.

2. Related Work

In this section, we examine the breadth of data transformation methods used in machine learning, before turning our attention to the specific domain in which we are interested: image representations of time series data. Finally, we introduce a toy problem, artifact detection within EEG data, which forms a testbed for comparative analysis of these image representations.

2.1. Alternative Data Representations

Alternative representations of raw data are produced by various data transformation pipelines. Data augmentation is a well-known application of these representations; augmented datasets bolster accuracy and mitigate overfitting [1].

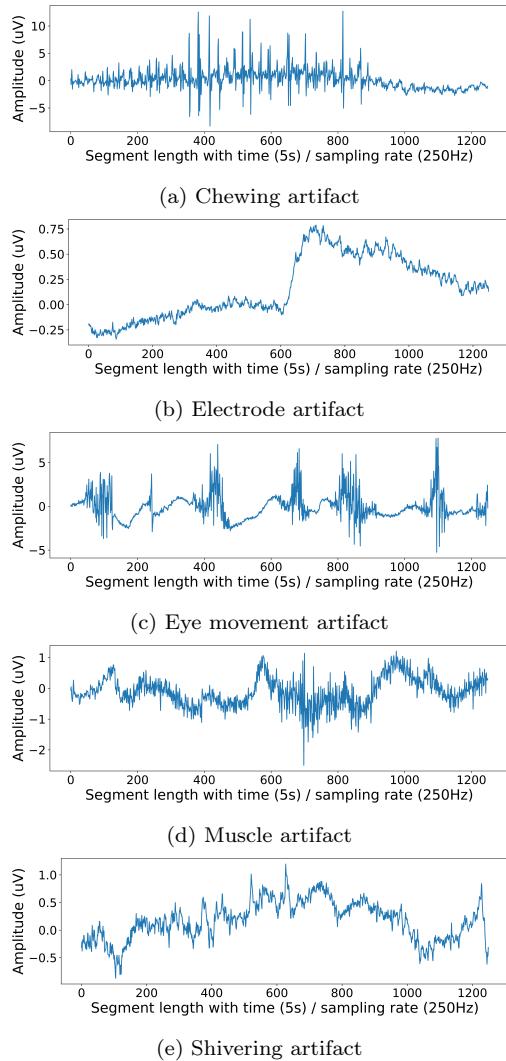


Figure 2: Different artifacts in EEG data.

Alternative representations may also make latent information in raw data amenable to downstream machine learning. In early computer vision research, filter-based feature extraction was essential to tree-based model understanding of image data [2]. Similarly, early work in speech processing represented voice data as mel-frequency cepstra to enable auditory information processing [3]. More recently, methods like node2vec have been developed to convert graph-based data into vectors for consumption by language models [4]. Similarly, word2vec, and other word embedding engines, represent words as vectors for neural language models [5].

2.2. *Image Representations of Time Series Data*

Many signal processing methods, such as short Fourier transforms, mel-spectrograms, cochleograms, and continuous wavelet transforms, have been used to convert sequential time series data into an image for visualization and analysis.

Individual image representations of time series data have been used with success in various domains, including speech [6], sensor data [7], and EEG data [8].

However, in this diverse landscape of image representations for sequence data, there is no clear winner. While many studies have shown the utility of transforming time series data into images, to our best knowledge, the field lacks comprehensive comparative analyses between these time series representation methods.

2.3. *Artifact Detection in EEG data*

In order to make strides towards such a comparative analysis, it is useful to introduce a toy task: artifact detection within EEG data.

Electroencephalography (EEG) is a valuable non-invasive method for recording the electrical activity of the brain. EEG offers high temporal resolution, capturing brain activity on the order of milliseconds [9]. However, EEG recordings often suffer from noise artifacts, irrelevant signals originating from various sources, including eye movements, muscle activity, and external interference [10]. Detecting and removing these artifacts is crucial for accurate EEG data analysis.

EEG artifact detection serves as a useful toy problem for studying data representation techniques [11].

Firstly, EEG artifact detection shares common characteristics with real-world problems involving time-series data [12]. Much like financial data,

EEG data is periodic, but noisy. Much like sensor data, EEG data involves multichannel measurements at high temporal resolution. And much like speech recognition data, EEG data contains information rich across the frequency spectrum. Thus, we consider EEG data to be broadly representative of many time series classification tasks.

Secondly, EEG artifact detection is a well-explored research area in neuroscience. Researchers have successfully applied machine learning techniques, including deep learning models pretrained on unrelated datasets like ImageNet, to enhance EEG data analysis [13]. Convolutional neural networks (CNNs) have proven effective in transforming EEG data into visual representations, improving artifact detection [14, 15]. This toy problem is still challenging, but tractable, making it useful for such a comparative analysis of data representation methods.

3. Methods

3.1. Dataset

We train and evaluate our classifier on the TUH EEG Artifact Corpus (TUAR) from the Temple University Hospital of Philadelphia (TUH), a subset of the TUH EEG Corpus [16]. The TUAR contains normal EEG signals, and EEG signals are affected by five types of artifacts: chewing events, eye movements, muscular artifacts, shivering events, and instrumental artifacts (such as electrode pop, electrostatic artifacts, or lead artifacts). These artifacts are subject to significant data imbalance; eye movement and muscle artifacts account for 45.7% and 35.9%, respectively. Electrode pops make up 15.9%, while chewing and shivering are the rarest artifacts with 2.2% and 0.1%, respectively (see Figure 2). The majority of the data however consists of normal EEG signals. In total, the dataset comprises 259 EEG sessions collected from 213 patients between 10 and 90 years old, over a span of 13 years. The EEGs were recorded with a sampling frequency of 250 Hz and 16-bit resolution.

3.2. Preprocessing

To preprocess the data, we segment the continuous EEG recordings into overlapping windows of EEG data. We chose a window length of 5 seconds based on previous research [17]. Since not all recordings had the same frequency, we downsampled them to a frequency of 128Hz. We included all

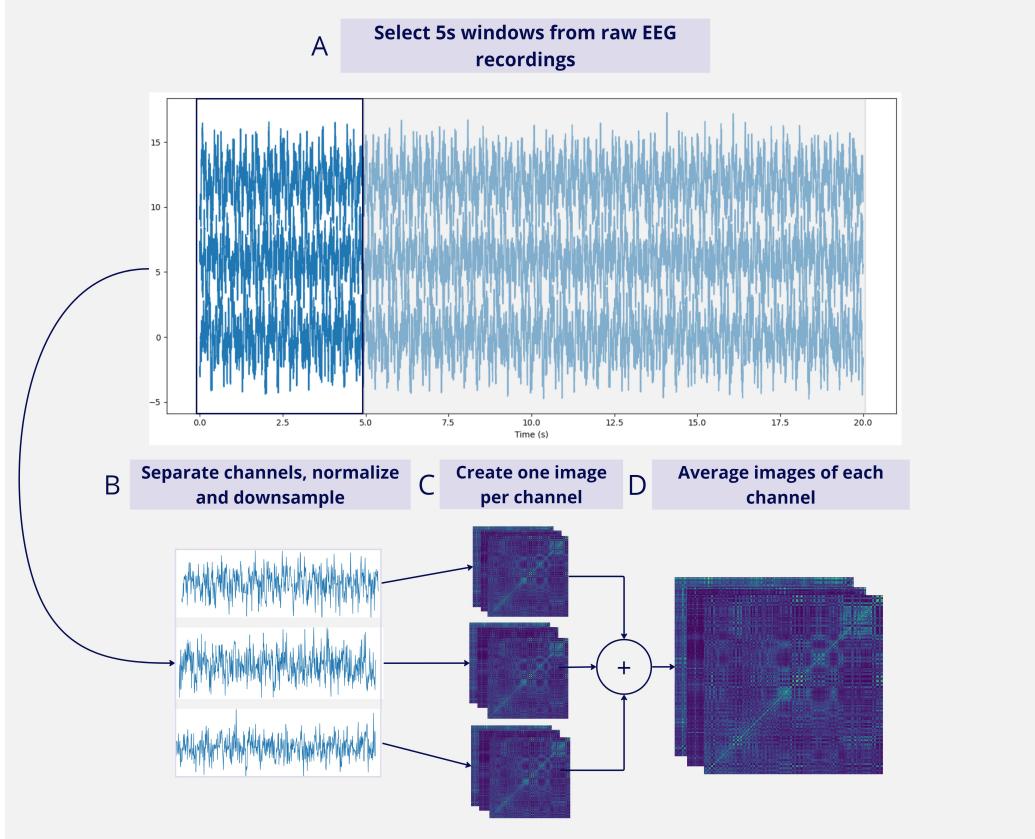


Figure 3: Raw EEG data undergoes four preprocessing steps: (a) partitioning into 5s windows, (b) channel separation, normalization, and downsampling to 128 Hz, (c) creating visual representations for each channel, and (d) averaging them into a single image representing all channels (except for correlation matrices, which ensemble channels by default). For the purpose of illustration, we depict only 3 channels and 3 representation techniques.

and only the channels that were present in all of the data, which amounted to 19 of over 50 channels. In line with common deep learning practice, we normalized the data with a z -transform (see Figure 3).

3.3. Time Series Data Representations

We selected a number of data representation methods which are often used in the literature [11, 18, 19]. Below, we introduce and define each method.

3.3.1. Correlation Matrices

Correlation matrices capture cross-channel similarities in multi-channel time series. We note that it is the only representation we examine that distills all channels into a single representation.

Formally, the correlation matrix C for n channels is defined as

$$C = \begin{bmatrix} \rho(S_1, S_1) & \cdots & \rho(S_1, S_n) \\ \vdots & \ddots & \vdots \\ \rho(S_n, S_1) & \cdots & \rho(S_n, S_n) \end{bmatrix} \quad (1)$$

where $\rho(S_i, S_j)$ is the correlation between channel S_i and S_j .

The correlation matrix C is visualized as a contour plot (see Figure 1a). Correlation matrices have been successfully used to capture spatial dependencies and relationships between different electrodes [8].

3.3.2. Recurrence Plots

Recurrence plots are simple method to identify recurrent sequences in time series data [19]. Given a time series $X = (x_1, \dots, x_N)$, the binarized recurrence matrix R is given by

$$R = \begin{bmatrix} \mathbf{1}(|x_1 - x_N| > \epsilon) & \cdots & \mathbf{1}(|x_1 - x_1| > \epsilon) \\ \vdots & \ddots & \vdots \\ \mathbf{1}(|x_N - x_N| > \epsilon) & \cdots & \mathbf{1}(|x_N - x_1| > \epsilon) \end{bmatrix} \quad (2)$$

where ϵ is some binarization threshold value, and $\mathbf{1}(x)$ is the Heaviside step function, which takes the value of 1 when the condition x is true, and 0 otherwise.

Thus, the recurrence plot R is a binary matrix where a value of 0 at position (i, j) indicates that data points x_i and x_j are considered recurrent (see Figure 1b). These plots are commonly used [20, 21] for EEG analysis, as normal EEG signals are consistently periodic, while abnormalities are aperiodic.

3.3.3. Gramian Angular Summation Fields

Gramian Angular Summation Fields (GASF) encode time series data into matrices that preserve temporal correlation information [22]. In this representation, each data point $x_t \in \{x_1, x_2, \dots, x_N\}$ at time t is mapped to a

polar coordinate with a mapping function ϕ , given by:

$$\phi(x_t) = \left(\frac{t}{N}, \arccos(x_t) \right) \quad (3)$$

Then, the GASF G is composed of the pairwise inner products of the angular representations of each data point:

$$G = \begin{bmatrix} \cos(\phi(x_1) + \phi(x_1)) & \dots & \cos(\phi(x_1) + \phi(x_N)) \\ \vdots & \ddots & \vdots \\ \cos(\phi(x_N) + \phi(x_1)) & \dots & \cos(\phi(x_N) + \phi(x_N)) \end{bmatrix} \quad (4)$$

The resulting GASF G is a 2D representation of the time series that captures its temporal correlations (see Figure 1c). These GASFs have proven to be successful in teasing out temporal trends in EEG data [23, 24].

3.3.4. Markov Transition Fields

Markov Transition Fields (MTF) are a matrix of transition probabilities between states of a time series [25]. That is, given a time series $X = \{x_1, x_2, \dots, x_N\}$, we divide the data into Q quantiles, and then assign each x_i to its corresponding bin $q \in [1, Q]$. Then, let us denote the bin containing x_i to be q_i .

The Markov Transition Field M is an $N \times N$ time-aware transition matrix, wherein each element M_{ij} represents the transition probability from the quantiles $q_i \rightarrow q_j$, containing x_i, x_j respectively. Thus, the MTF is given by

$$M = \begin{bmatrix} v_{11|x_1 \in q_1, x_1 \in q_1} & \dots & v_{1N|x_1 \in q_1, x_N \in q_N} \\ \vdots & \ddots & \vdots \\ v_{N1|x_N \in q_N, x_1 \in q_1} & \dots & v_{NN|x_N \in q_N, x_N \in q_N} \end{bmatrix} \quad (5)$$

Where each value v_{ij} represents the one-step transition probability from $q_i \rightarrow q_j$ in the time series.

The resulting figure visualizes this matrix as a heatmap (see Figure 1d). These fields have been previously used to model temporal dependencies within electroencephalogram (EEG) data [26].

3.3.5. Continuous Wavelet Transforms

The Continuous Wavelet Transform (CWT) allows us to visualize how the frequency components of a time series change over time [27].

Formally, the Continuous Wavelet Transform (CWT) of a signal $s(t)$ is defined as:

$$\text{CWT}(s(t))(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} s(t)\psi^* \left(\frac{t-b}{a} \right) dt \quad (6)$$

where $\text{CWT}(s(t))(a, b)$ represents the CWT of the signal $s(t)$ with respect to the parameters a and b , where a is the scaling parameter controlling the width of the wavelet function, b is the translation parameter shifting the wavelet function along the time axis, and $\psi^*(t)$ is the complex conjugate of the mother wavelet function $\psi(t)$ used for the transformation.

The resulting plot represents time on the horizontal axis and frequency on the vertical axis. The intensity of colors or shading in the representation indicates the magnitude or power of the different frequencies at each time point. Brighter or more intense colors represent higher power, indicating the presence of a stronger frequency component (See Figure 1e). Prior work has found success in representing EEG data with CWTs [28, 29].

3.3.6. Spectograms

We generate spectrograms from Short-Time Fourier Transforms (STFT). Like CWT, STFT is a time-frequency analysis technique [30]. In contrast to CWT, STFT uses fixed windows, and offers either good frequency or time resolution, while CWT uses variable-width wavelets, providing better time-frequency localization and adaptability to non-stationary signals.

Formally, the STFT of a signal $s(t)$ is defined as:

$$\text{STFT}\{s(t)\}(\tau, \omega) = \int_{-\infty}^{\infty} s(t)w(t-\tau)e^{-i\omega t}dt \quad (7)$$

where $w(\tau)$ is the window function and ω is the frequency. This formula essentially applies a Fourier transform to "windows" of the signal $x(t)$, where each window is obtained by multiplying $x(t)$ by the window function $w(t)$. The window function is usually chosen to be zero outside a certain interval so that the integral is over a finite range.

A spectrogram is an intensity plot of an STFT (see Figure 1f). Spectrograms are commonly used to represent EEG data [31, 32].

4. Experiments

We evaluated our six representations with eleven well-known deep learning architectures. Here, we detail the architectures and experimental procedure used.

4.1. Models and Finetuning

We utilized a selection of CNNs pre-trained on the ImageNet dataset, accessible through the Keras Applications library, as the basis for our models. ImageNet is a large-scale dataset of annotated images that is widely used in computer vision research and machine learning [41]. We decided to fine-tune 11 models according to their parameter count (P): four large models (with $P \geq 56M$), four medium-sized models (with $20M < P < 56M$), and three relatively small models ($P < 20M$). For an overview of the models we fine-tuned see table 1. We trained the models with binary cross-entropy loss to predict the presence of a specific artifact.

The labels are encoded as one-hot vectors. Details of the training process can be found in Appendix B.

5. Results

The highest performance was achieved with markov transition fields (MTF), closely followed by correlation matrices and spectrograms.

Xception and EfficientNetB0 emerged as frontrunners in this evaluation. Notably, Xception demonstrated an F1-score of 90.5% coupled with 89% accuracy. EfficientNet B0 showed comparable performance achieving an F1-score of 89.4% and a test accuracy of 89.9% (see Figure 4).

F1-scores varied widely across models when MTFs were employed. The superior model exhibited an F1-score that was 12% higher than the worst-performing model on MTFs. This discrepancy is noteworthy as it is twice the average difference observed between the best and worst-performing models in each testing condition. In comparison to F1, accuracy varied much less with the best model achieving at most 6% higher scores than others for a given visualization technique.

For both metrics, the majority of performance differences between visualization techniques were statistically significant at a 5% significance level, as

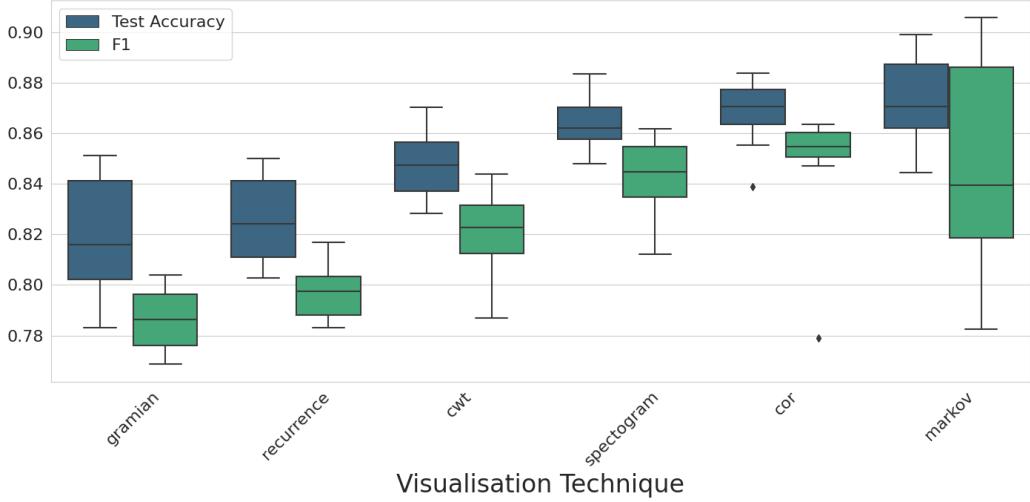


Figure 4: Accuracy and F1 scores on the validation set for each visualization technique, averaged across all models.

determined by a paired t-test. Details can be found in Appendix A.

6. Discussion

Our representations are a classic example of the bias-variance trade-off [42]. Some of our representations, such as that of correlation matrices and recurrence plots, contain high bias, and correspondingly, models on these representations fit faster, with a lesser risk of overfitting. Meanwhile, more expressive representations, such as those derived from Fourier transforms, are invertible, and so do not remove any semantic information. However, these representations faithfully represent the noise within the dataset, making downstream models susceptible to overfitting.

It is telling that our top representations lie within the happy medium; in particular, Markov transition matrices are a fairly information-rich representation that, while not fully invertible, was also quite resistant to overfitting.

6.1. Practitioner’s Recommendations

Within the representations we consider, spectrograms and continuous wavelet transform represent high variance data, while correlation matrices and recurrence plots represent high bias high bias representations.

In truth, the ideal representation for any given use case will depend on the signal-to-noise ratio of the data. However, within the regimes of low, moderate, and high bias representations, we recommend spectrograms, Markov transition matrices, and correlation matrices, respectively. In line with previous work [8, 43] on these representations, they appear to outperform similar-complexity representations by virtue of capturing and highlighting more salient features within underlying data.

6.2. Limitations

This work does not claim to have done a comprehensive assessment of data representation methods; instead, we make strides towards a more nuanced understanding of various image representations of time series data.

As such, it is important to highlight several limitations of our work:

- We explored 6 methods for representing sequential data as images, choosing those which we found to be the most popular in prior work. However, there are many such representations which we did not explore, such as mel-spectrograms and cochleograms, which may be promising.
- Similarly, we attempted to make our analysis data representations independent of downstream model choice by profiling performance across 11 well-known CNN model architectures, ranging from 3.5 million to 138.4 million parameters. However, we don't expect our findings to generalize to models outside of CNNs, and similarly, to models much smaller and larger than our explorations.
- Finally, while we believe that our choice of a toy problem is useful insofar that it is broadly representative of in-the-wild time series data, we recognize that generalization across domains is a challenging task, and that our findings are unlikely to generalize to data which looks diametrically different from EEG data.

6.3. Future Work

We believe this work acts as a starting point towards a nuanced understanding of data representation strategies for use in deep learning. Promising directions for future work may include:

- Expanding the set of representations considered, and the types and sizes of models considered.

- Further comparative analyses on non-image-based representations of time series data, or data representations deriving from non-sequential data.
- Considering other toy problems which differ dramatically from EEG data, i.e. time series data without periodic behavior, or with information concentrated within a small band of frequencies.

7. Conclusion

In this work, we evaluate six image representations of sequential data within a toy problem, artifact detection within EEGs. Three representations, namely Markov Transition Fields, Correlation Matrices, and Spectrograms, have consistently demonstrated superior performance. Given the utility of such representations in improving the performance of downstream machine learning models, we are optimistic that further research in this direction will be useful to the field writ large.

8. Acknowledgements/Notes

All authors declare that they have no conflicts of interest.

References

- [1] C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, *Journal of big data* 6 (1) (2019) 1–48.
- [2] K. He, J. Sun, X. Tang, Guided image filtering, *IEEE transactions on pattern analysis and machine intelligence* 35 (6) (2012) 1397–1409.
- [3] S. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, *IEEE transactions on acoustics, speech, and signal processing* 28 (4) (1980) 357–366.
- [4] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.
- [5] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781* (2013).
- [6] T. Arias-Vergara, P. Klumpp, J. C. Vasquez-Correa, E. Nöth, J. R. Orozco-Arroyave, M. Schuster, Multi-channel spectrograms for speech processing applications using deep learning methods, *Pattern Analysis and Applications* 24 (2021) 423–431.
- [7] C.-L. Yang, C.-Y. Yang, Z.-X. Chen, N.-W. Lo, Multivariate time series data transformation for convolutional neural network, in: *2019 IEEE/SICE International Symposium on System Integration (SII)*, IEEE, 2019, pp. 188–192.
- [8] N. Bahador, K. Erikson, J. Laurila, J. Koskenkari, T. Ala-Kokko, J. Koertelainen, A correlation-driven mapping for deep learning application in detecting artifacts within the eeg, *Journal of Neural Engineering* 17 (5) (2020) 056018.
- [9] C. M. Michel, D. Brunet, EEG source imaging: A practical review of the analysis steps 10.
URL <https://www.frontiersin.org/articles/10.3389/fneur.2019.00325>

- [10] X. Jiang, G.-B. Bian, Z. Tian, Removal of artifacts from EEG signals: A review 19 (5) 987. doi:10.3390/s19050987.
URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6427454/>
- [11] H. Tiwary, A. Bhavsar, Time-frequency representations for eeg artifact classification with cnns, in: 2021 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), IEEE, 2021, pp. 1–8.
- [12] T.-c. Fu, A review on time series data mining, Engineering Applications of Artificial Intelligence 24 (1) (2011) 164–181.
- [13] M. T. Sadiq, M. Z. Aziz, A. Almogren, A. Yousaf, S. Siuly, A. U. Rehman, Exploiting pretrained cnn models for the development of an eeg-based robust bci framework, Computers in Biology and Medicine 143 (2022) 105242.
- [14] G. Xu, X. Shen, S. Chen, Y. Zong, C. Zhang, H. Yue, M. Liu, F. Chen, W. Che, A deep transfer convolutional neural network framework for eeg signal classification, IEEE Access 7 (2019) 112767–112776.
- [15] N. Bahador, K. Erikson, J. Laurila, J. Koskenkari, T. Ala-Kokko, J. Koertelainen, Automatic detection of artifacts in EEG by combining deep learning and histogram contour processing 138–141Conference Name: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) in conjunction with the 43rd Annual Conference of the Canadian Medical and Biological Engineering Society ISBN: 9781728119908 Place: Montreal, QC, Canada Publisher: IEEE. doi:10.1109/EMBC44109.2020.9175711.
URL <https://ieeexplore.ieee.org/document/9175711/>
- [16] A. Hamid, K. Gagliano, S. Rahman, N. Tulin, V. Tchiong, I. Obeid, J. Picone, The temple university artifact corpus: An annotated corpus of EEG artifacts, in: 2020 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), pp. 1–4, ISSN: 2473-716X. doi:10.1109/SPMB50085.2020.9353647.
- [17] W. Y. Peh, Y. Yao, J. Dauwels, Transformer convolutional neural networks for automated artifact detection in scalp EEG. arXiv: 2208.02405 [eess].
URL <http://arxiv.org/abs/2208.02405>

- [18] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.-A. Muller, Deep learning for time series classification: a review, *Data mining and knowledge discovery* 33 (4) (2019) 917–963.
- [19] K. Nakano, B. Chakraborty, Effect of data representation for time series classification—a comparative study and a new proposal, *Machine Learning and Knowledge Extraction* 1 (4) (2019) 1100–1120.
- [20] G. Ouyang, X. Li, C. Dang, D. A. Richards, Using recurrence plot for determinism analysis of eeg recordings in genetic absence epilepsy rats, *Clinical neurophysiology* 119 (8) (2008) 1747–1755.
- [21] F. Bahari, A. Janghorbani, Eeg-based emotion recognition using recurrence plot analysis and k nearest neighbor classifier, in: 2013 20th Iranian Conference on Biomedical Engineering (ICBME), IEEE, 2013, pp. 228–233.
- [22] Z. Wang, T. Oates, et al., Encoding time series as images for visual inspection and classification using tiled convolutional neural networks, in: Workshops at the twenty-ninth AAAI conference on artificial intelligence, Vol. 1, AAAI Menlo Park, CA, USA, 2015.
- [23] M. M. Islam, M. M. H. Shuvo, Densenet based speech imagery eeg signal classification using gramian angular field, in: 2019 5th International Conference on Advances in Electrical Engineering (ICAEE), IEEE, 2019, pp. 149–154.
- [24] K. P. Thanaraj, B. Parvathavarthini, U. J. Tanik, V. Rajinikanth, S. Kadry, K. Kamalanand, Implementation of deep neural networks to classify eeg signals using gramian angular summation field for epilepsy diagnosis, *arXiv preprint arXiv:2003.04534* (2020).
- [25] Z. Wang, T. Oates, Spatially encoding temporal correlations to classify temporal data using convolutional neural networks, *arXiv preprint arXiv:1509.07481* (2015).
- [26] A. Shankar, S. Dandapat, S. Barma, Discrimination of types of seizure using brain rhythms based on markov transition field and deep learning, *IEEE Open Journal of Instrumentation and Measurement* 1 (2022) 1–8.

- [27] A. Grossmann, J. Morlet, Decomposition of hardy functions into square integrable wavelets of constant shape, SIAM journal on mathematical analysis 15 (4) (1984) 723–736.
- [28] Z. Khademi, F. Ebrahimi, H. M. Kordy, A transfer learning-based cnn and lstm hybrid deep learning model to classify motor imagery eeg signals, Computers in biology and medicine 143 (2022) 105288.
- [29] A. Narin, Detection of focal and non-focal epileptic seizure using continuous wavelet transform-based scalogram images and pre-trained deep neural networks, Irbm 43 (1) (2022) 22–31.
- [30] J. Allen, Short term spectral analysis, synthesis, and modification by discrete fourier transform, IEEE Transactions on Acoustics, Speech, and Signal Processing 25 (3) (1977) 235–238.
- [31] G. Ruffini, D. Ibañez, M. Castellano, L. Dubreuil-Vall, A. Soria-Frisch, R. Postuma, J.-F. Gagnon, J. Montplaisir, Deep learning with EEG spectrograms in rapid eye movement behavior disorder 10.
URL <https://www.frontiersin.org/articles/10.3389/fneur.2019.00806>
- [32] S. P. Kyathanahally, A. Döring, R. Kreis, Deep learning approaches for detection and removal of ghosting artifacts in MR spectroscopy 80 (3) 851–863, _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/mrm.27096>.
doi:10.1002/mrm.27096.
URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.27096>
- [33] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
- [34] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105–6114.
- [35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

- [36] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 31, 2017.
- [37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
- [38] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.
- [39] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
- [40] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).
- [41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255, ISSN: 1063-6919. doi:10.1109/CVPR.2009.5206848.
- [42] B. Neal, On the bias-variance tradeoff: Textbooks need an update, arXiv preprint arXiv:1912.08286 (2019).
- [43] M. Jalayer, C. Orsenigo, C. Vercellis, Fault detection and diagnosis for rotating machinery: A model based on convolutional lstm, fast fourier and continuous wavelet transforms, Computers in Industry 125 (2021) 103378.
- [44] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

Appendix A. Full Results

visualization	Model	Accuracy	Precision	Recall	F1
cor	densenet201	0.855	0.850	0.844	0.847
	efficientnetb0*	0.875	0.863	0.860	0.862
	efficientnetb7*	0.875	0.853	0.868	0.861
	inceptionresnetv2*	0.884	0.861	0.866	0.864
	inceptionv3	0.879	0.842	0.866	0.854
	mobilenet	0.871	0.846	0.864	0.855
	mobilenetv2	0.868	0.855	0.846	0.850
	resnet152	0.860	0.847	0.855	0.851
	resnet50	0.867	0.846	0.865	0.855
	vgg16	0.839	0.839	0.727	0.779
cwt	densenet201*	0.870	0.861	0.828	0.844
	efficientnetb0*	0.862	0.855	0.832	0.843
	efficientnetb7	0.849	0.834	0.823	0.828
	inceptionresnetv2	0.832	0.791	0.830	0.810
	inceptionv3	0.842	0.835	0.800	0.817
	mobilenet	0.843	0.830	0.816	0.823
	mobilenetv2	0.865	0.854	0.816	0.835
	resnet152	0.828	0.818	0.801	0.809
	resnet50	0.850	0.839	0.809	0.824
	vgg16	0.847	0.847	0.734	0.787
gramian	densenet201	0.847	0.832	0.765	0.797
	efficientnetb0	0.807	0.806	0.782	0.794
	efficientnetb7*	0.839	0.836	0.794	0.815
	inceptionresnetv2*	0.851	0.839	0.771	0.804
	inceptionv3	0.804	0.800	0.752	0.775
	mobilenet	0.783	0.773	0.765	0.769
	mobilenetv2	0.825	0.819	0.779	0.799
	resnet152	0.831	0.822	0.762	0.791
	resnet50	0.802	0.794	0.763	0.778
	vgg16*	0.845	0.842	0.730	0.782

visualization	Model	Accuracy	Precision	Recall	F1
markov	densenet201	0.857	0.854	0.779	0.814
	efficientnetb0**	0.899	0.903	0.886	0.894
	efficientnetb7	0.898	0.900	0.890	0.895
	inceptionresnetv2	0.881	0.880	0.839	0.859
	inceptionv3	0.861	0.858	0.780	0.817
	mobilenet	0.884	0.887	0.869	0.878
	mobilenetv2	0.871	0.872	0.809	0.839
	resnet152	0.863	0.865	0.780	0.820
	resnet50	0.864	0.866	0.813	0.839
	vgg16	0.844	0.844	0.729	0.782
	xception**	0.891	0.896	0.916	0.906
recurrence	densenet201	0.850	0.839	0.786	0.811
	efficientnetb0	0.817	0.806	0.799	0.802
	efficientnetb7*	0.855	0.854	0.814	0.834
	inceptionresnetv2	0.835	0.829	0.780	0.804
	inceptionv3	0.826	0.819	0.777	0.798
	mobilenet	0.822	0.799	0.795	0.797
	mobilenetv2	0.846	0.840	0.795	0.817
	resnet152	0.803	0.802	0.771	0.787
	resnet50	0.804	0.789	0.791	0.790
	vgg16	0.843	0.843	0.731	0.783
	xception	0.809	0.770	0.805	0.787
spectrogram	densenet201*	0.883	0.812	0.854	0.833
	efficientnetb0*	0.870	0.869	0.855	0.862
	efficientnetb7*	0.871	0.859	0.860	0.860
	inceptionresnetv2	0.860	0.819	0.855	0.837
	inceptionv3	0.848	0.845	0.835	0.840
	mobilenet	0.862	0.852	0.852	0.852
	mobilenetv2*	0.875	0.877	0.839	0.858
	resnet152	0.862	0.853	0.837	0.845
	vgg16	0.849	0.822	0.843	0.833
	xception	0.855	0.856	0.772	0.812

Table A.2: The performance of each combination of model and data representation. Models that perform best on a single metric for each representation are highlighted with *. Models that perform best on a single metric overall are highlighted with **.

Appendix A.1. Comparing Performance across Models

Here, we examine the effect of different architectures on performance, focusing on Markov Transition Fields (see Figure A.5). For this representation, Xception and EfficientNet models performed best, both in terms of accuracy and F1 score.

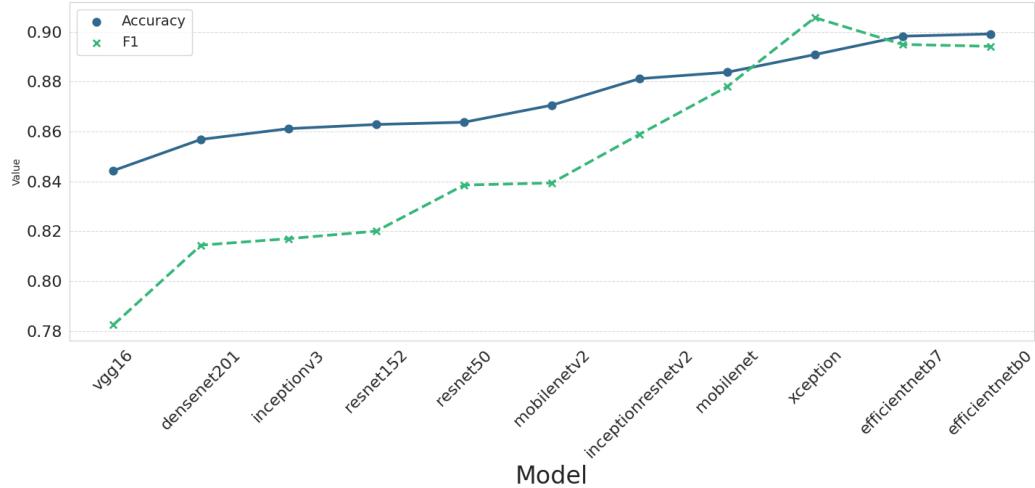


Figure A.5: Comparing the performance of different architectures on MTFs.

Next, we look at the performance differences between models when averaged over all data representations. Some models' performance was much more sensitive to data representations than others'; VGG16 was remarkably consistent in its performance, while Xception was 10% more accurate on some representations than on others (see Figure A.6).

Appendix A.2. Comparing Performance across Data Representations

The majority of average performance differences in accuracy and F1 scores between visualization techniques were statistically significant at a 5% significance level, as determined by a paired t-test (see Figure A.7). We found four combinations were the difference was not statistically significant: MTFs and spectrograms, MTFs and correlation matrices, correlation matrices and spectrograms and, lastly recurrence plots and GASFs.

Appendix B. Training details

Each pretrained model was modified in two ways: first, we prepend a convolutional layer to convert the image representations into image features,

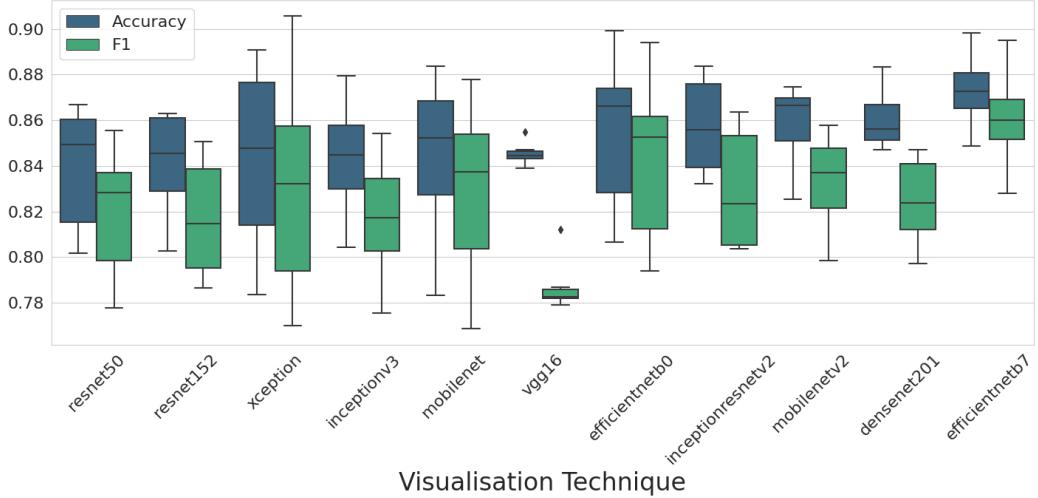


Figure A.6: Comparing F1 and Accuracy across models, averaged over visualization techniques.

and secondly, we replace the ImageNet classification layer with a global average pooling layer and a 5-node fully-connected layer with sigmoid activation for our classification task.

Our training process involves a training stage and a fine-tuning stage. In the training stage, we keep the pretrained model backbone frozen, and only train the added layers for 5 epochs, utilizing the Adam optimizer [44] with a learning rate of 0.001. In the finetuning stage, we unfreeze all the pretrained model layers except the batch normalization layers, and continue training until convergence with a learning rate of 0.00001.

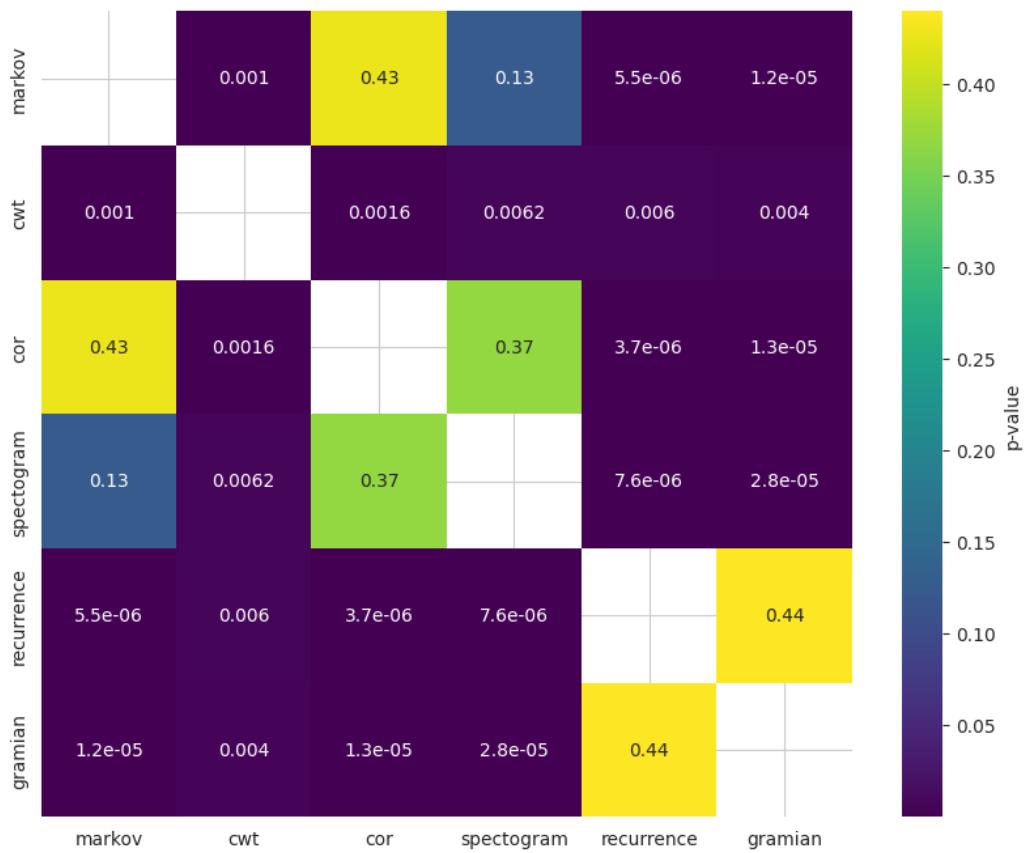


Figure A.7: Assessing the statistical significance of average differences in accuracy between visualization techniques. The analysis was performed with a paired t-test.