A Hypergraph-Based Alternative to Convolutional Layer in Image Classification Models

Eronides Felisberto da Silva Neto and Juliano B. Lima

***Abstract*— Hypergraph modeling has emerged as an effective approach for representing data by capturing higher-order relationships among multiple data instances. Recent methods utilizing hypergraphs demonstrate applications in signal processing such as image denoising, compression and spectral clustering. To perform learning tasks on hypergraph representation of image data, hypergraph neural networks have been introduced to enhance performance. This paper explores the effectiveness of hypergraph signal representations as input features for neural network-based image classification models. The results show that a mixed hypergraph network achieves performance comparable to traditional convolutional neural networks, while requiring less computation, as evidenced by shorter training time.**

***Keywords*— Hypergraph, signal processing, image classification**

I. INTRODUCTION

Large volumes of data (big data) generated by Internet of Things (IoT) devices, social media interactions, and traffic patterns can exhibit complex structures, making it challenging to apply traditional data analysis techniques [1]. In this context, graph-based representations have emerged as a powerful paradigm for modeling and interpreting such data. Graphs provide a flexible and expressive framework capable of capturing diverse relationships, including device interactions, network topologies, and information flows, thereby facilitating the identification of meaningful patterns and insights [2].

While graphs enable the modeling of novel structures and relationships among device connections—often resulting in matrix-based analyses—algebraic operations within these representations become inefficient when capturing interactions beyond pairwise connections. To address the limitations of traditional graphs in modeling higher-order interactions (HOIs), hypergraphs have been introduced as a more expressive framework for representing complex signal interactions [3], [4].

In the context of data generated by IoT applications, hypergraphs offer a powerful framework for modeling data structures such as time series, images, and multidimensional arrays [5]. By employing tensor-based representations, hypergraphs enable the analysis of multi-relational data, capturing refined relationships and interactions that extend beyond pairwise relationships. To facilitate efficient data and spectral analysis, hypergraph signal processing (HGSP) frameworks [3], [4], [6] extends classical signal processing tools to accommodate the rich and intricate relationships captured by hypergraphs.

Recent research has also explored the intersection between HGSP operations and the development of hypergraph neural networks (HGNN) architectures, aiming to leverage the spectral properties of hypergraphs to enable learning tasks [7]. Furthermore, recent surveys have provided comprehensive overviews of HGNN architectures, training methodologies, and their diverse applications [8].

Considering the potential of hypergraph modeling for representing images [9] by incorporating spatial information, recent research has explored the integration of such representations in various image processing tasks, including image classification [10] and segmentation [11]. This approach establishes a foundation for applying hypergraph-based representations in deep learning frameworks. In particular, HGNNs can offer a promising avenue for analyzing and improving the performance of image classification models by incorporating HOIs within the structured data.

In this paper, we propose a hypergraph signal representations as input features for neural network-based image classification models. Specifically, we assess the performance of two mixed HGNN architectures on the MNIST and Fashion-MNIST (FMNIST) datasets. The proposed approach is benchmarked against conventional convolutional neural network (CNN) models to compare classification performance and highlight the potential advantages of incorporating hypergraph-based representations.

This paper is organized as follows. In Section II, we summarize the theoretical foundations of hypergraphs, HGSP based on tensor operations and HGNNs. Section III describes the methodology proposed to the evaluation of mixed HGNNs and CNNs. In section IV, we discuss the results of the evaluation. Finally, Section V brings the conclusions.

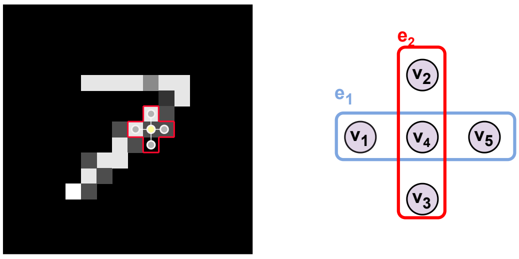
II. PRELIMINARIES

A. Hypergrah Basics

In a nutshell, a undirected hypergraph is defined as a pair $\mathcal{H} = (\mathcal{V}, \varepsilon)$, where $\mathcal{V} = \{v_1, ..., v_N\}$ is a set of nodes and $\varepsilon = \{e_1, ..., e_K\}$ a set of hyperedges. The concept of hyperedges extends the edge concept, being possible to connect more than two nodes (vertices), capturing relationships involving more than two entities. The parameter that summarizes the difference between a graph and a hypergraph is the maximum cardinality of a hyperedge, defined as $M = m.c.e.(\mathcal{H}) = \max\{|e_i| : e_i \in \varepsilon\}$, being a graph a special case when $M = 2$. Another particular case is the k-uniform hypergraph, where each hyperedge connects exactly k vertices, meaning all hyperedges have the same cardinality. Figure 1 illustrates the process of constructing a hypergraph from an image.

The mathematical representation of hypergraphs are generally representend by an incidence matrix $\mathbf{H} \in \{0, 1\}^{|\mathcal{V}| \times |\mathcal{E}|}$. Extending the concept of graphs, and considering the higher

Eronides F. da Silva Neto and Juliano B. Lima, Department of Electronics and Systems, Federal University of Pernambuco (UFPE), Recife-PE, Brazil. E-mails: eronides.silvaneto@ufpe.br; juliano.lima@ufpe.br.



a) b)

Fig. 1. a) illustrate the representation of an image by a hypergraph based on each pixel being a node and b) the hyperedges creation based on proximity of pixels neighborhood.

order interactions, a k-uniform hypergraph $\mathcal{H} = (\mathcal{V}, \varepsilon)$ can also be represented by an adjacency tensor, considering N nodes and M, being represented by an M-th order, N-dimensional adjacency tensor $\mathcal{A} \in \mathbb{R}^{N^M}$. The last generalization from graph theory is the concept of Laplacian tensor, also applied exclusively to k-uniform hypergraphs. Considering $\mathcal{D}$ as an M-th order N-dimensional super-diagonal tensor with non-zero entries, calculated from the degrees of $\mathbf{H}$, the Laplacian tensor is defined as

$$\mathcal{L} = \mathcal{D} - \mathcal{A} \in \mathbb{R}^{N^M}. \tag{1}$$

Different from adjacency tensor that encodes the presence of hyperedges for structural modeling, the Laplacian tensor provides an way to encode the hypergraph structure and spectral properties.

B. Hypergraph Signal Processing

Hypergraph signal processing extends graph signal processing (GSP) by capturing higher-order relationships beyond pairwise interactions. Among the earliest studies, Zhang et al. [4] proposed an HGSP framework with applications in spectral clustering, data compression, and signal denoising. While theoretically grounded and supported by practical examples, the approach has limitations—most notably, its reliance on the orthogonal decomposition of the adjacency tensor via the CANDECOMP/PARAFAC (CP) method [12], which lacks an exact solution.

As an alternative, Pena-Pena et al. [6] proposed a novel HGSP framework based on the tensor-tensor product (t-product) algebra, which has demonstrated effectiveness in preserving the intrinsic structure of tensor data. Considering two third-order tensors, $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ and $\mathcal{B} \in \mathbb{R}^{N_2 \times N_4 \times N_3}$, the t-product is defined by

$$\mathcal{C} = \mathcal{A} * \mathcal{B}, \tag{2}$$

with $\mathcal{C} \in \mathbb{R}^{N_1 \times N_4 \times N_3}$ calculated as

$$\mathcal{C} = \text{fold}(\text{bcirc}(\mathcal{A}) \cdot \text{unfold}(\mathcal{B})). \tag{3}$$

The operator `bcirc`($\mathcal{A}$) maps the frontal slices of the tensor $\mathcal{A}$ into a block circulant matrix. Similarly, the `unfold`($\mathcal{B}$) operator stacks the frontal slices of the tensor $\mathcal{B}$ vertically, resulting in a matrix of dimensions $N_2N_3 \times N_4$. The `fold()` and `unfold()` operators are mutually inverse. Given that the core principles of HGSP are founded on the t-product algebra, we now proceed with the relevant definitions.

Hypergraph signal: considering the framework [6], a possible representation of a hypergraph signal from a one dimensional signal is defined as an $(M-1)$-order, N-dimensional tensor $\mathcal{X}$, computed as the outer product of the original signal on the hypergraph $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix}^T$, that is

$$\mathcal{X} = \mathbf{x} \circ \dots \circ \mathbf{x}. \tag{4}$$

With the objective of representating in higher-order dimensions, the hypergraph signal can be expanded to a second dimension and adjusted to a symmetric version to be integrated to the tensor product framework. This is represented by

$$\overrightarrow{\mathcal{X}}_s = \text{sym}(\text{expand}(\mathcal{X})), \tag{5}$$

with $\overrightarrow{\mathcal{X}}_s \in \mathbb{R}^{N \times 1 \times 2N+1}$. In addition to one-dimensional signals, hypergraph signals can be also created from a set of signals [6]. Considering a signal $\mathbf{X} \in \mathbb{R}^{N \times L}$, it is possible to associate L one-dimensional signals with the hypergraph in a direct way.

Hypergraph shifiting and filtering are fundamental operations in signal processing, enabling key functionalities such as time delay, sampling, signal reconstruction, and convolution. Analogously to its role in GSP, the shift operation in HGSP ($\mathcal{F}_s$) is generalized to encompass any operator that captures the relational dependencies among nodes in a hypergraph. In this context, both the adjacency tensor $\mathcal{A}$ and the Laplacian tensor $\mathcal{L}$ can be employed as shift operators.

Based on the previous definitions for a hypergraph signal (5), the one-time filtered hypergraph signal is calculated by

$$\overrightarrow{\mathcal{Y}}_s = \mathcal{F}_s * \overrightarrow{\mathcal{X}}_s. \tag{6}$$

Hypergraph filtering extends hypergraph shifting by manipulating hypergraph signals via linear transformations. The filtering operation follows the same formulation as (6), with the shifting operator replaced by the filter representation $\mathcal{Q}_s$.

C. Hypergrah neural networks

Convolutional neural networks are a class of deep learning algorithms inspired by the organization of the human brain and the concept of artificial neurons. CNNs consist of multiple layers of interconnected units that automatically and adaptively learn spatial hierarchies of features from input data. Through the use of convolutional operations, they effectively capture local patterns, making them particularly well-suited for tasks involving image and signal processing. Mathematically, a neural network can be represented as a mathematical function that maps a set of inputs to an output [13],

$$y = f\left(x, \theta\right), \tag{7}$$

where θ denotes the set of trainable parameters, comprising both weights and biases.

Hypergraph neural networks are a class of neural networks designed to perform learning tasks on data represented as hypergraphs, enabling the modeling of HOIs. Considering the variety of data structured as hypergraphs, various HGNNs approaches are designed to meet specific applications requirements [10], [14], [15]. According to Wang et al. [16], a HGNN learns a representation mapping f_θ that combines node features $\mathbf{X}$ and the hypergraph descriptor, utilizing either a tensor shifting operator $\mathcal{F}$ or matrix-based approaches, such as the incidence matrix $\mathbf{H}$.

With the purpose of working with tensor based operations, the direct analogue representation of hypergraph computation operations, based on HGNNs is represented by

$$y = f_\theta \left(\mathcal{X}_s, \mathcal{F}_s \right) \tag{8}$$

where f_θ is a mathematical function based on calculation and modeling based on vertices or hyperedges.

From the perspective of HGSP operations, [7] provides a unified view of the HGNN architecture, which is constructed around two fundamental operations: signal transformation and signal shifting. Additionally, the work introduces an HGNN architecture that is equivalent to a tensor-based hypergraph signal denoising operation. In addition to approaches that rely solely on tensor-based operations and representations, some network proposals use weighted fusion hypergraphs and regular neural network layers operations [14].

Leveraging the flexibility of hypergraph representations for diverse data types, HGNN architectures construction allows the integration of HGSP operations with conventional convolutional neural network layers within a unified architecture, enabling more flexible and powerful architectures.

III. Evaluating Image Classification Models based on Hypergraph Signals

The application of classification models to images is a fundamental task in the field of computer vision, involving the assignment of a semantic label to an image based on its visual characteristics. This paper proposes the use of hypergraph signal representations as input features for image classification models based on neural networks. Specifically, the proposed approach involves constructing a mixed HGNN that integrates hypergraph structural representations with traditional neural network layers, replacing conventional convolutional layers with hypergraph-based signal processing and representation. To evaluate the proposed mixed HGNN with tensor based hypergraph signal, accuracy is adopted as the performance metric due to its intuitive interpretation and ease of comparison across models.

Datasets. Standard benchmark datasets for grayscale image classification, such as MNIST, and FMNIST, are employed to evaluate and conduct a comparative analysis of image classification models, using the proposed mixed HGNN approach. The MNIST dataset comprises grayscale images of handwritten digits and is commonly used as a foundational benchmark for evaluating basic image classification algorithms. In contrast, the Fashion-MNIST dataset offers a more challenging alternative by presenting grayscale images of various clothing items, thereby introducing greater intra-class variability.

Hypergraph signal creation. Instead of using the original image representation as the input of neural networks, this work proposes a representation of each image as a hypergraph signal defined within the t-product framework. Each image is first resized to 16×16 pixels and then partitioned into four non-overlapping 4×4 blocks. For each of these blocks, a hypergraph can be constructed following the methodology presented in [9], where each pixel is treated as a node. As a result, each block yields an associated hypergraph and a corresponding hypergraph signal. The hypergraph signal is defined as a one-dimensional array of length 16, under the assumption that this signal exhibits smoothness over the hypergraph structure (with $N = 16$) derived from the respective image block.

For each original image from the respective dataset, represented as hypergraphs, a hypergraph signal $\overrightarrow{\mathcal{X}_s} \in \mathbb{R}^{16\times1\times33}$ is associated as a representation of high-order interactions. Compared to both the original image representation and hypergraph signal formulations based on $N = 256$, the approach proposed in this work, using a hypergraph constructed from 4×4 image blocks ($N = 16$), offers improved computational efficiency while maintaining meaningful structural information.

Mixed hypergraph neural network architectures. To evaluate the performance of hypergraph signal representations as feature vectors within neural network architectures, this study proposes two distinct mixed HGNN architectures:

- **Simple mixed HGNN (SM-HGNN)**: a network that receives the hypergraph signal ($16 \times 1 \times 33$) as input, the first layer is a channel expansion, followed by a 2×2 max-pooling layer, followed by a batch normalization layer. The feature maps are then flattened and passed through two fully connected layers, the first with 256 units followed by 128 units and dropout (rate = 0.1), before a final softmax output layer for classification into 10 categories.
- **Simple mixed HGNN with denoising (SMD-HGNN)**: as same as the previous network, but with a denoising layer before the channel expansion, as shown on Fig. 2.

The filtering operation incorporated in SMD-HGNN involves a denoising operation implemented within the t-product framework. Despite the datasets used in this evaluation are relatively clean, some imagens can still exhibit pixel-level variations or artifacts due to preprocessing. As introduced in the previous section, the equation for a filtering process is exactly the same as (6) with the operator replaced by the filter representation $\mathcal{Q}_s$, which is constructed through hard-thresholding of the spectral coefficients λ_c

$$\mathcal{Q}_s[l,l,k] = \begin{cases} 1, \text{if } \lambda_l \leq \lambda_c \\ 0, \text{otherwise}, \end{cases} \tag{9}$$

where $l = 1, 2, ..., N$ and $k = 1, 2, ..., 2N + 1$. The cut-off frequency used in the architecture design is $\lambda_c = 0.10 \times N$.

Experimental setup and benchmarks evaluation with CNNs. The construction and evaluation of each neural network model follow a holdout validation procedure, where the dataset is randomly partitioned into 80% for training and 20%

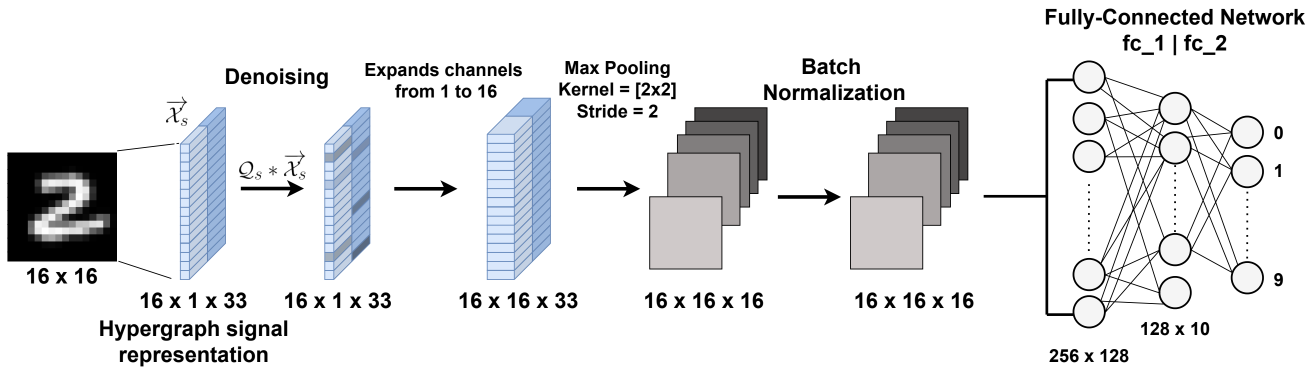


Fig. 2. Architecture overview of the proposed mixed HGNN with hypergraph signal denoising.

for testing. The supervised training is conducted using the backpropagation algorithm over 10 epochs, optimized using the Adam optimizer with a learning rate of 0.01. A softmax activation function is applied at the output layer, and the negative log-likelihood loss is used as the training criterion. To ensure the robustness and reliability of the results, this process is repeated across 35 independent runs, each with a distinct random seed uniformly sampled between 0 and 8675309. This repeated evaluation allows for assessing the consistency of the model's performance under varying data splits.

As a benchmark dataset for image classification, MNIST provides up-to-date comparisons of state-of-the-art methods, facilitating performance evaluation across a wide range of models [17]. The CNN models chosen for evaluation in this work, with layers detailed on Table I, were selected based on their architectural implementation with the inclusion of convolutional layers, which are essential for a meaningful comparison with the proposed HOIs representation derived from hypergraph signal representation. Additionally, both selected CNN architectures are well-suited for image classification tasks on the MNIST and FMNIST [18] datasets, ensuring consistency and comparability across experiments.

TABLE I

LAYER COMPARISON BETWEEN 1-LAYER CNN AND VGG16 H-CNN

Model	Layers	Model	Layers
1-layer CNN	Convolutional Layer (kernel=3, padding=1) MaxPool2d (2x2) Fully Connected 1 Fully Connected 2	**VGG16 H-CNN**	Conv Layer (3x3, pad=1) ReLU Conv Layer (3x3, pad=1) ReLU MaxPool2d (2x2) **(Repeating this block three times)** Fully Connected 1 Fully Connected 2

All CNNs and mixed HGNNs were implemented in Python (version 3.10.12) using the PyTorch deep learning framework (version 2.6). The experiments were conducted on a machine equipped with an 11th Gen Intel® Core™ i5-1135G7 processor running at 2.40GHz with 8 logical cores. This setup provided sufficient computational resources for training and evaluating the proposed models efficiently. In addition to the accuracy metric, the training time is recorded for each run.

IV. Results and Discussion

Table II summarizes the experimental results obtained for the evaluated models on the selected datasets, presenting the classification accuracy for each configuration. As shown in Table II, VGG16 H-CNN [19], a hierarchical CNN designed for image classification, featuring a consistent 16-layer structure with convolutional blocks followed by max-pooling for spatial reduction, achieved the highest average accuracy among all evaluated models for both datasets, while SM-HGNN demonstrated a result close to the best model, with lower computational complexity based on the required training time to create the model. For the MNIST dataset, a direct comparison between the 1-layer convolutional CNN and the SMD-HGNN yields comparable results in terms of accuracy. However, the SMD-HGNN demonstrates a clear advantage in computational efficiency, requiring on average approximately 35.33 seconds to construct the model, compared to 966.56 seconds for VGG16.

To determine whether statistically significant differences in accuracy exist among the four models evaluated for each dataset, the Kruskal–Wallis hypothesis test was applied [20]. This non-parametric method is appropriate for comparing accuracy distributions across multiple independent groups, particularly when the assumption of normality is not satisfied. The results for the MNIST dataset indicated the presence of a model whose performance differed significantly from that of the other models.

To identify which specific models differed significantly and which exhibited statistically equivalent performance, a post-hoc Dunn's test with Bonferroni correction was subsequently performed. The results of the pairwise comparisons confirmed that the VGG16 H-CNN model outperformed the others with statistically significant differences in both datasets. Moreover, no significant difference was found between the CNN and SMD-HGNN models, indicating statistically equivalent performance between them.

Applying the same methodology to the FMNIST dataset revealed that the pairwise comparisons indicated all models were statistically different from one another. Unlike MNIST, the denoising layer SMD-HGNN failed to significantly improve the model, likely due to the increased visual complexity and finer-grained object details present in FMNIST images, which

TABLE II
SUMMARY OF CLASSIFICATION ACCURACY AND TRAINNING TIME FOR EVALUATED MODELS

Model	MNIST		Fashion MNIST	
	Accuracy	Training Time (s)	Accuracy	Training Time (s)
CNN	0.9812 ± 0.0017	54.97 ± 4.24	0.8957 ± 0.0041	70.67 ± 4.60
VGG16 H-CNN	0.9882 ± 0.0028	966.56 ± 23.53	0.9089 ± 0.0039	936.93 ± 30.67
SM-HGNN	0.9770 ± 0.0016	30.18 ± 1.30	0.8813 ± 0.0034	31.5349 ± 0.20
SMD-HGNN	0.9813 ± 0.0014	35.33 ± 5.23	0.8874 ± 0.0036	33.8883 ± 3.19

may limit the effectiveness of the smoothing mechanism.

V. CONCLUSION AND FUTURE WORKS

The most significant finding of this study is the demonstration that a traditional CNN with a convolutional layer can be effectively replaced by a network with a hypergraph signal representation based on the t-product framework. From a broader perspective, considering both computational effort and classification accuracy, the results indicate that the SMD-HGNN architecture achieves performance comparable to that of a purely convolutional model. This result is achieved with greater computational efficiency, as the model does not rely on traditional convolutional layers. These findings highlight the potential of hypergraph-based approaches as viable alternatives in image classification tasks.

Regarding the network architecture proposed by the mixed HGNNs, the input data are represented using the t-product HGSP framework, which enables block circulant interactions across slices of the tensor and captures multi-dimensional, higher-order relationships that are not accessible through conventional matrix-based transformations [6]. Despite the demonstrated effectiveness of this hypergraph signal representation, further refinement of signal transformation techniques may enhance the performance of the proposed architectures and support the adaptation of additional CNN models for extended evaluation. Moreover, the development of specialized transformation and shifting blocks within HGNNs could enable the construction of equivalent representations for more complex convolutional structures, facilitating a more comprehensive and rigorous comparison across different network architectures.

As a continuation of this work, future research could focus on the design of a specialized HGNN architecture built entirely on tensor operations derived from the t-product HGSP framework. This approach would further refine the current evaluation and open new possibilities for improving performance. Moreover, enhancing the methodology for hypergraph construction from image data and exploring more sophisticated transformation and shifting operations could provide deeper insights into optimizing mixed architectures for image classification tasks.

ACKNOWLEDGMENTS

The authors wish to thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) under grants 442238/2023-1, 312935/2023-4, and 405903/2023-5, and by the Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE) under grant APQ-1226-3.04/22.

REFERENCES

[1] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[2] G. B. Ribeiro and J. B. Lima, "Graph signal processing in a nutshell," *Journal of Communication and Information systems*, vol. 33, no. 1, 2018.

[3] S. Barbarossa and M. Tsitsvero, "An introduction to hypergraph signal processing," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6425–6429, March 2016.

[4] S. Zhang, Z. Ding, and S. Cui, "Introducing hypergraph signal processing: Theoretical foundation and practical applications," *IEEE Internet of Things Journal*, vol. 7, pp. 639–660, Jan 2020.

[5] Q. Dai and Y. Gao, *Hypergraph Computation*. Springer Nature, 2023.

[6] K. Pena-Pena, D. L. Lau, and G. R. Arce, "t-hgsp: Hypergraph signal processing using t-product tensor decompositions," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 9, pp. 329–345, 2023.

[7] F. Wang, K. Pena-Pena, W. Qian, and G. R. Arce, "A unified view between tensor hypergraph neural networks and signal denoising," in *2023 31st European Signal Processing Conference (EUSIPCO)*, pp. 1968–1972, IEEE, 2023.

[8] S. Kim, S. Y. Lee, Y. Gao, A. Antelmi, M. Polato, and K. Shin, "A survey on hypergraph neural networks: An in-depth and step-by-step guide," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6534–6544, 2024.

[9] A. Bretto and L. Gillibert, "Hypergraph-based image representation," in *Graph-Based Representations in Pattern Recognition* (L. Brun and M. Vento, eds.), (Berlin, Heidelberg), pp. 1–11, Springer Berlin Heidelberg, 2005.

[10] Z. Ma, Z. Jiang, and H. Zhang, "Hyperspectral image classification using feature fusion hypergraph convolution neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[11] S. Zhang, S. Cui, and Z. Ding, "Hypergraph-based image processing," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 216–220, 2020.

[12] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[14] X. Hao, J. Li, M. Ma, J. Qin, D. Zhang, and F. Liu, "Hypergraph convolutional network for longitudinal data analysis in alzheimer's disease," *Computers in Biology and Medicine*, vol. 168, p. 107765, 2024.

[15] Y. Rong, L. Nong, Z. Liang, Z. Huang, J. Peng, and Y. Huang, "Hypergraph position attention convolution networks for 3d point cloud segmentation," *Applied Sciences*, 2024.

[16] F. Wang, K. Pena-Pena, W. Qian, and G. R. Arce, "T-hypergnns: Hypergraph neural networks via tensor representations," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[17] "Mnist benchmark (image classification)." https://paperswithcode.com/sota/image-classification-on-mnist. Accessed: 2025-05-03.

[18] Y. Seo and K. shik Shin, "Hierarchical convolutional neural networks for fashion image classification," *Expert Systems with Applications*, vol. 116, pp. 328–339, 2019.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[20] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, p. 1–30, Dec. 2006.