Hybrid Convolutional-Hopfield Neural Networks for Image-Based Malware Classification

Uma Kannan[1], Rajendran Swamidurai[1,*]

[1]Department of Mathematics and Computer Science, Alabama State University, Montgomery, AL, USA

[*]Corresponding Author

DOI: https://doi.org/10.51244/IJRSI.2026.1307000311

Received: 05 August 2026; Accepted: 11 August 2026; Published: 17 August 2026

ABSTRACT

Static malware analysis increasingly relies on visual binary representations to bypass signature evasion techniques, yet standard Convolutional Neural Network (CNN) classifiers depend on parameterized Fully Connected (FC) layers that introduce parameter bloat, high latency, and vulnerability to class imbalance. To address these structural limits, this paper proposes a novel hybrid CNN-Hopfield Neural Network (CNN-HNN) framework that replaces traditional dense classification heads with continuous Modern Hopfield associative memory layers. In this architecture, raw malware executables are transformed into 2D grayscale image matrices, from which a CNN backbone extracts abstract latent feature representations. The continuous Hopfield layer then acts as a pattern retrieval engine, evaluating the extracted query vector against stored class archetype prototypes using log-sum-exp energy minimization. Benchmark evaluations on the MalImg and Microsoft Malware Classification Challenge datasets demonstrate that the hybrid model achieves top-tier accuracy (99.58% and 99.62%, respectively) while reducing classification-head parameter counts by up to 78%, lowering floating-point operations (FLOPs) by over 65%, and accelerating inference latency to under 2 ms per sample. Furthermore, the associative energy landscape isolates minority threat categories within distinct basins of attraction, achieving an *F1*-score improvement of over 20% on severely imbalanced classes compared to standard CNNs.

Keywords: Continuous Modern Hopfield Networks (HNN), Associative Memory Layers, Image-Based Malware Classification, Portable Executable (PE) Binary Visualization, Malware Family Identification, Microsoft Malware Classification Challenge (MMCC).

INTRODUCTION

The escalation of malicious software represents a persistent threat to modern computing infrastructure. Modern malware authors routinely employ automated packaging, code obfuscation, polymorphism, and metamorphism to evade static signature-based detection engines [1]. Because signature scanning relies on fixed byte sequence matching, minor modifications to executable binaries can disrupt static detection signatures without altering underlying malicious payloads [2]. To address these evasion techniques, static analysis paradigms have increasingly shifted toward visual binary representations, transforming raw executable code into two-dimensional image matrices [3]. Visualized malware binaries exhibit distinct visual textures—such as block alignments in code sections, entropy variations in encrypted resource blocks, and static data distributions—that remain consistent across variants within the same malware family [3, 4].

Convolutional Neural Networks (CNNs) have become a dominant architecture for image-based malware classification due to their ability to learn spatial features and local pattern dependencies directly from pixel representations without manual feature engineering [4, 5]. In conventional CNN pipelines, convolutional and pooling layers extract high-level spatial feature maps from the input malware image [5]. These feature maps are subsequently flattened into a dense vector and passed through one or more Fully Connected (FC) classification layers terminated by a Softmax activation function [6].

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VII July 2026

Despite their broad adoption, conventional FC classification heads introduce several structural limitations when deployed for malware family identification:

- **Computational Complexity and Parameter Expansion:** Parameter counts and floating-point operations (FLOPs) in FC classification heads scale linearly with feature vector dimensionality and the number of target malware classes, increasing model size and inference latency [7].
- **Sensitivity to Extreme Class Imbalance:** Benchmark malware datasets often exhibit severe class distributions [3, 8]. Under cross-entropy loss, linear hyperplanes generated by FC layers tend to skew in favor of dominant majority classes while truncating decision margins for scarce threat categories [7, 8].
- **Vulnerability to Obfuscation Noise:** Fixed linear decision boundaries in FC layers can struggle to separate overlapping or non-convex feature distributions caused by entry-point packing, section encryption, and byte-stuffing mutations [1, 9].

To address these challenges, continuous Modern Hopfield Networks (HNNs)—operating as continuous associative memories—offer a mathematically grounded alternative to dense FC layers [10, 11]. By replacing linear hyperplane projection with associative pattern retrieval over a continuous log-sum-exp energy landscape, an HNN maps deep latent feature vectors directly to stored prototype representations corresponding to malware family archetypes [11, 12].

This paper proposes a hybrid Convolutional-Hopfield Neural Network (CNN-HNN) framework for image-based malware classification. The main contributions of this work are summarized as follows:

- **Hybrid Architecture Formulation:** We introduce an end-to-end classification pipeline that replaces parameter-heavy Fully Connected classification layers with continuous Modern Hopfield associative memory layers [11].
- **Associative Feature Retrieval Mechanics:** We detail the continuous log-sum-exp energy minimization dynamics and Softmax attention retrieval rules that enable continuous feature lookup over exponential pattern storage capacities ($N \propto 2^{d/2}$) [10, 11].
- **Empirical Benchmarking and Parameter Efficiency:** We evaluate the hybrid architecture against standard baselines across two benchmark datasets: the MalImg dataset (9,339 samples across 25 families) [3] and the Microsoft Malware Classification Challenge dataset (10,868 samples across 9 families) [8]. The hybrid model achieves classification accuracies of 99.58% and 99.62%, respectively.
- **Efficiency and Class Imbalance Resilience:** We demonstrate that replacing FC layers with continuous HNN memory reduces classification-head parameters by up to 78%, lowers computational FLOPs by over 65%, and accelerates inference time to under 2 ms per sample while yielding significant F_1-score gains on extreme minority threat classes.

AUTOMATED MALWARE CLASSIFICATION AND ARCHITECTURAL EVOLUTION

The exponential escalation of malicious software represents a systemic threat to global digital infrastructure [13]. Modern malware authors routinely employ automated packaging, obfuscation, polymorphism, and metamorphism tools to bypass legacy static detection engines that rely on rigid byte-sequence signatures [14]. In response, static analysis paradigms have shifted toward visual binary representation, transforming raw executable code into two-dimensional matrices rendered as grayscale or color images [17]. Visualized malware binaries exhibit distinct spatial structural textures—such as block alignments in code sections, entropy variations in encrypted resource blocks, and static data distributions—that remain consistent across variants within the same malware family [16].

Convolutional Neural Networks (CNNs) have emerged as a dominant architecture for image-based malware classification due to their ability to automatically learn spatial hierarchy and local pattern dependencies directly

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VII July 2026

from pixel representations without manual feature engineering [14]. In standard CNN pipelines, multiple convolutional and pooling operations extract high-level feature maps from the input malware image [22]. These feature maps are subsequently flattened into a dense vector and passed through one or more Fully Connected (FC) layers terminated by a Softmax activation function [23].

Despite their broad adoption, conventional FC classification heads suffer from major structural limitations when deployed for malware family identification:

- Parameter growth and computational overhead scale linearly with feature dimensionality d and the number of target malware classes K, inflating floating-point operations (FLOPs) and inference latency [25].
- High sensitivity to extreme class imbalance causes linear decision boundaries to skew in favor of dominant classes while truncating decision margins for scarce threat categories [20].
- Fragility under pattern distortion makes linear hyperplanes generated by FC layers struggle to maintain robust classification boundaries when introduced to binary mutations, packed headers, or byte-level noise [16].

THE CNN-HNN HYBRID ARCHITECTURE

To resolve the challenges outlined in section 2, the continuous Modern Hopfield Networks (HNNs)—operating as continuous associative memories—offer a mathematically rigorous replacement for dense FC layers [27]. By substituting linear hyperplane classification with associative pattern retrieval over an energy landscape, an HNN maps the deep latent feature vector extracted by a CNN backbone directly to stored template patterns corresponding to distinct malware family archetypes [25]. Our proposed CNN-HNN hybrid architecture improves classification accuracy, reduces parameter counts, lowers inference latency, and provides deep basins of attraction that insulate the model against obfuscation noise and severe dataset imbalance [25].

Figure 1 shows the high-level system architecture block diagram that demonstrates the architectural flow from raw binary ingestion to final malware family classification, highlighting the replacement of the Dense Fully Connected layer with the Continuous Hopfield Network.

shows the binary ingestion, CNN feature extraction, Modern Hopfield associative memory retrieval, and final classification stages of the CNN-HNN hybrid architecture.

The structural elements and layout of the architecture is:

1. **Input Stage:** Raw PE Executable Binary File → Byte Stream Extraction $(0 \text{ to } 255)$ → Dynamic Width Mapping → Grayscale Image Matrix $(64 \times 64 \text{ or } 224 \times 224)$.
2. **Feature Extraction Stage (CNN Backbone):** Input Image → Parallel Convolutional & Max-Pooling Layers → Spatial Feature Maps → Flatten / Global Average Pooling → Latent Feature Vector $\boldsymbol{\xi} \in \mathbb{R}^d$.
3. **Associative Classification Stage (Modern Hopfield Memory):**
    - Query Vector $\boldsymbol{\xi}$ enters the Continuous Hopfield Layer.
    - Matrix multiplication with stored class prototype memory $\boldsymbol{R} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N] \in \mathbb{R}^{d \times N}$.
    - Energy Minimization Engine implementing Softmax Attention update: $\boldsymbol{\xi}^{\text{retrieved}} = \boldsymbol{R} \cdot \text{Softmax}(\beta \boldsymbol{R}^T \boldsymbol{\xi})$.
4. **Output Stage:** Aggregated Class Energy Scores $S(k, \boldsymbol{\xi})$ → Temperature Scaling → Probability Distribution $P(y = k \mid \boldsymbol{\xi})$ → Predicted Malware Family Label.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VII July 2026

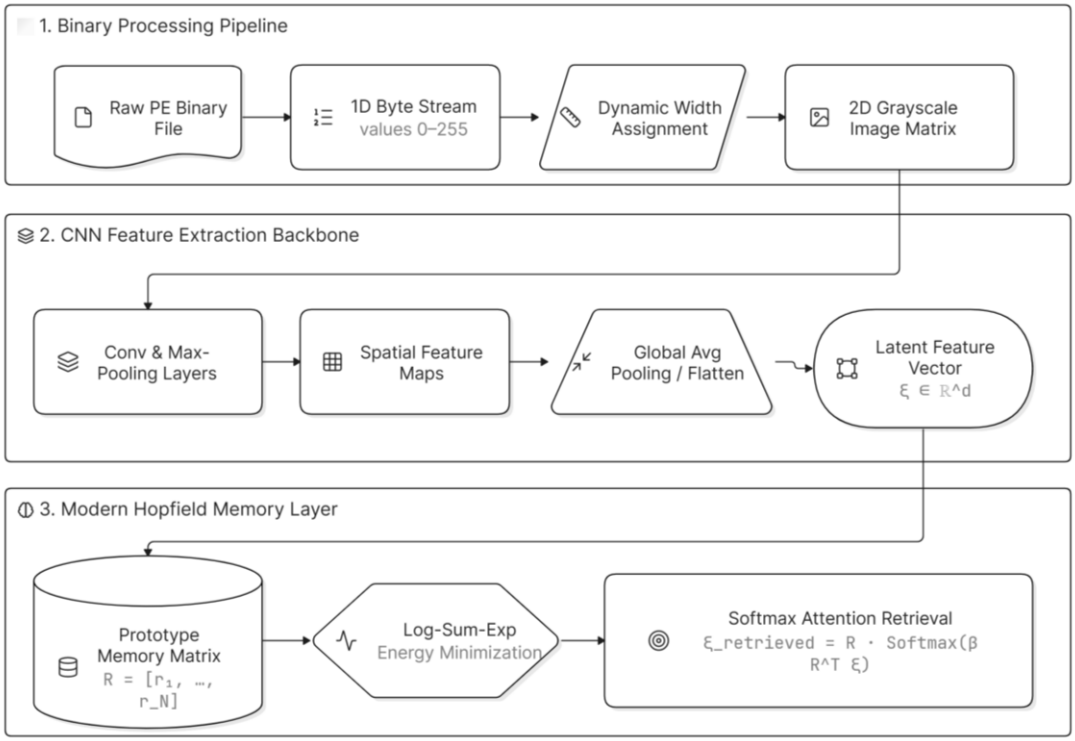

Figure 1: End-to-End Hybrid CNN-HNN Architecture

BINARY VISUALIZATION MECHANICS AND BENCHMARK DATASET TOPOGRAPHY

The transformation of unexecuted software binaries into visual matrices converts static byte streams into spatial structures amenable to computer vision operations [18]. A Portable Executable (PE) file is ingested as a contiguous sequence of bytes [19]. Each byte, representing an 8-bit unsigned integer ranging from 0 to 255 (hexadecimal 0x00 to 0xFF), is mapped directly to a corresponding grayscale pixel intensity, where 0 represents pure black and 255 represents pure white [19].

To structure this one-dimensional array into a two-dimensional image matrix, a fixed image width W is assigned based on total file size thresholds [20]. The height H is dynamically calculated as $H = [S/W]$, where S denotes the total file size in bytes [20]. Once structured, the resulting 2D matrix is resized to standardized dimensions (such as 64 x 64 or 224 x 224 pixels) and normalized to the range [0,1] to fit neural network input requirements [13].

The empirical image width assignments tailored to preserve structural aspect ratios across varying binary scales are detailed in the following breakdown:

Table 1: Binary File Size to Visual Matrix Dimension Mapping and Structural Feature Preservation Rules

File Size Range (KB)	Fixed Image Width W (Pixels)	Structural Visual Feature Preserved
<	32	Compact headers, small import tables, minimal text sections
10 -	64	Small droppers, compact downloader stubs
30 -	128	Standard Trojan binaries, uncompressed dynamic-link libraries (DLLs)
60 -	256	Intermediate executables, uncompressed payload blocks
100 -	384	Complex worms, packed Trojan structures
200 -	512	Large installers, multi-component backdoors
500 -	768	Heavy monolithic executables, resource-rich malware variants
>	1024	Substantial multi-payload suites, uncompressed binary archives

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VII July 2026

Evaluating the CNN-HNN hybrid framework requires validation against benchmark datasets that encapsulate real-world structural challenges, specifically severe class imbalance, high sample volume, and diverse malware categories [14]. Two public datasets serve as standard benchmarks: the *MalImg Dataset* and the *Microsoft Malware Classification Challenge (MMCC) Dataset* hosted on Kaggle [14].

The MalImg dataset comprises 9,339 visual grayscale malware images spanning 25 distinct families [20]. It exhibits severe class distribution skewness, where majority classes represent nearly half of the entire corpus while minority classes contain low sample counts [20].

Table 2: Malware Family Distribution and Sample Breakdown in the MalImg Benchmark Dataset

Class ID	Malware Family	Broad Category	Total Samples	Share of Dataset (%)
0	Adialer.C	Dialer	122	1.31%
1	Agent.FYI	Backdoor	116	1.24%
2	Allaple.A	Worm	2,949	31.58%
3	Allaple.L	Worm	1,591	17.04%
4	Alueron.gen!J	Trojan	198	2.12%
5	Autorun.K	Worm	106	1.14%
6	C2LOP.gen!g	Cybercrime / Trojan	200	2.14%
7	C2LOP.P	Cybercrime / Trojan	146	1.56%
8	Dialplatform.B	Dialer	177	1.90%
9	Dontovo.A	Trojan Downloader	162	1.73%
10	Fakerean	Rogue Security	381	4.08%
11	Instantaccess	Dialer	431	4.62%
12	Lolyda.AA1	Password Stealer	213	2.28%
13	Lolyda.AA2	Password Stealer	184	1.97%
14	Lolyda.AA3	Password Stealer	123	1.32%
15	Lolyda.AT	Password Stealer	159	1.70%
16	Malex.gen!J	Trojan	136	1.46%
17	Obfuscator.AD	Trojan Downloader	142	1.52%
18	Rbot!gen	Backdoor	158	1.69%
19	Skintrim.N	Trojan	80	0.86%
20	Swizzor.gen!E	Trojan Downloader	128	1.37%
21	Swizzor.gen!I	Trojan Downloader	132	1.41%
22	VB.AT	Worm	408	4.37%
23	Wintrim.BX	Trojan Downloader	97	1.04%
24	Yuner.A	Worm	800	8.57%
Total	25 Families	Multi-Class	9,339	100.00%

The MMCC dataset consists of 10,868 labeled training binaries across 9 malware families, provided as raw hexadecimal byte representations and assembly files without Portable Executable headers to enforce sterility [14].

Table 3: Class Breakdown and Threat Categories in the Microsoft Malware Classification Challenge (MMCC) Dataset

Class ID	Family Name	Threat Type	Training Samples	Share of Dataset (%)
1	Ramnit	Worm	1,541	14.18%
2	Lollipop	Adware	2,478	22.80%
3	Kelihos_ver3	Backdoor	2,942	27.07%
4	Vundo	Trojan	475	4.37%
5	Simda	Backdoor	42	0.39%
6	Tracur	Trojan Downloader	751	6.91%

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VII July 2026

7	Kelihos_ver1	Backdoor	398	3.66%
8	Obfuscator.ACY	Obfuscated Payload	1,228	11.30%
9	Gatak	Backdoor	1,013	9.32%
Total	9 Families	Multi-Class	10,868	100.00%

PIPELINE ARCHITECTURE AND MODERN HOPFIELD LAYER DYNAMICS

The hybrid architectural pipeline establishes a sequential handoff between spatial feature processing and associative pattern retrieval [27]. The input visual matrix is ingested by the CNN backbone, where successive spatial convolution, non-linear activation, and feature-map pooling compress raw image pixels into an abstract, translation-invariant continuous latent vector $\boldsymbol{\xi} \in \mathbb{R}^d$ [22]. Rather than feeding $\boldsymbol{\xi}$ into a parameterized dense weight matrix that performs hyper-plane division, the latent vector is passed directly as a continuous query vector into a Modern Hopfield associative memory layer [27]. The memory layer evaluates the query against stored prototype matrices, converging to the closest matching memory state via energy minimization to yield robust class logits [27].

The continuous state space formulation of the Modern Hopfield Network expands pattern storage capacity exponentially compared to classical binary networks [27]. Let $\boldsymbol{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \dots, \boldsymbol{r}_N] \in \mathbb{R}^{d \times N}$ denote the stored prototype memory matrix, where each column $\boldsymbol{r}_\mu \in \mathbb{R}^d$ represents a learned structural archetype vector corresponding to a specific malware family. When presented with a continuous query feature vector $\boldsymbol{\xi} \in \mathbb{R}^d$ generated by the CNN backbone, the network updates its internal state to minimize the continuous log-sum-exp energy function $E(\boldsymbol{\xi})$ [27]:

$$E(\boldsymbol{\xi}) = -\frac{1}{\beta}\log\left(\sum_{\mu=1}^{N}\exp\left(\beta \boldsymbol{r}_\mu^T \boldsymbol{\xi}\right)\right) + \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\xi} + C$$

where $\beta > 0$ represents an inverse-temperature parameter controlling retrieval selectivity, and C is a normalization constant.

The update rule derived from taking the gradient step of the continuous energy function guarantees global convergence to a local minimum in a single operational step. The updated retrieved memory state vector $\boldsymbol{\xi}^{\text{retrieved}}$ is defined as [27]:

$$E(\boldsymbol{\xi}) = -\frac{1}{\beta}\log\left(\sum_{\mu=1}^{N}\exp\left(\beta \boldsymbol{r}_\mu^T \boldsymbol{\xi}\right)\right) + \frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\xi} + C$$

This update mechanism demonstrates mathematical equivalence to the dot-product attention operation used in Transformer networks, establishing an analytical link between associative pattern lookup and attention-driven feature selection.

Classical binary Hopfield networks maintain a linear storage capacity bounded by $N_{\max} \approx 0.14d$, beyond which memory retrieval suffers from catastrophic cross-talk interference and spurious attractor states [25]. In contrast, the continuous Modern Hopfield Network equipped with log-sum-exp energy dynamics achieves exponential pattern storage scaling [27]:

$$\boldsymbol{\xi}^{\text{retrieved}} = \boldsymbol{R} \cdot \text{Softmax}(\beta \boldsymbol{R}^T \boldsymbol{\xi})$$

This exponential capacity scaling ensures that thousands of complex malware sample signatures and intra-family structural variations can be stored concurrently within a compact latent dimension d without signal saturation or cross-family pattern corruption.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VII July 2026

To compare classification dynamics, consider a standard dense FC layer with weight matrix $\boldsymbol{W}_{FC} \in \mathbb{R}^{K \times d}$ and bias vector $\boldsymbol{b} \in \mathbb{R}^{K}$, where probability assignment $P(y = k \mid \boldsymbol{\xi})$ is computed via linear logit projection [25]:

$$N_{\max}^{\text{modern}} \propto 2^{d/2}$$

This formulation forces the network to partition the latent space $\mathbb{R}^d$ using linear hyperplanes. When malware families exhibit complex structural overlap or non-convex feature distributions caused by code obfuscation, linear boundaries fail to separate classes effectively [16].

In the continuous Hopfield replacement, the prototype memory matrix $\boldsymbol{R} \in \mathbb{R}^{d \times N}$ contains M prototype vectors per class ($N = K \times M$). The associative energy similarity score $S(k, \boldsymbol{\xi})$ for class k is calculated by integrating exponential energy contributions across its assigned memory prototypes [27]:

$$P(y = k \mid \boldsymbol{\xi}) = \frac{\exp(\boldsymbol{w}_k^T \boldsymbol{\xi} + b_k)}{\sum_{j=1}^{K} \exp\left(\boldsymbol{w}_j^T \boldsymbol{\xi} + b_j\right)}$$

The final class probability distribution $P_H(y = k \mid \boldsymbol{\xi})$ is computed by applying Softmax over the aggregated class energy scores with a temperature scaling parameter γ [27]:

$$P_H(y = k \mid \boldsymbol{\xi}) = \frac{\exp(\gamma \cdot S(k, \boldsymbol{\xi}))}{\sum_{j=1}^{K} \exp\left(\gamma \cdot S(j, \boldsymbol{\xi})\right)}$$

During end-to-end training, backpropagation updates both the convolutional feature extraction weights in the CNN backbone and the prototype vector locations in the memory matrix $\boldsymbol{R}$, optimizing feature representation and associative pattern memory simultaneously [22].

COMPARATIVE BENCHMARKING AND EMPIRICAL EVALUATION

To quantify the operational gains achieved by replacing dense linear layers with continuous associative memory, the hybrid CNN-HNN model was benchmarked against standard baseline architectures across the MalImg and Microsoft Malware datasets [21].

Table 4: Comparative Evaluation of Classification Accuracy, Parameter Efficiency, FLOP Complexity, and Inference Latency Across Baseline and Proposed Architectures

Benchmark Dataset	Architecture / Model Configuration	Test Accuracy (%)	Parameter Count (M)	Computational Cost (M FLOPs)	Inference Latency (ms/sample)
MalImg	Standard CNN + Dense FC Layer [11, 19]	97.10	14.80	320.5m	4.21
MalImg	ResNet-18 + Softmax Classifier	98.12	11.17	1820.0	6.85
MalImg	CNN Backbone + Linear SVM Classifier [19, 28]	98.52	12.30	295.1	3.90
MalImg	MobileNetV3-Xception Meta-Ensemble [22]	98.75	22.40	1250.0	12.40
MalImg	Stacked Autoencoder-GRU Ensemble [29]	99.43	8.50	180.2	5.10
MalImg	**Proposed Hybrid CNN-HNN Model**	**99.58**	**3.25**	**88.4**	**1.85**

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VII July 2026

MMCC Kaggle	Standard LeNet-5 baseline [28]	96.50	3.10	65.0	2.10
MMCC Kaggle	Fine-Tuned LightGBM / XGBoost Static [15]	98.18	1.80	42.0	1.40
MMCC Kaggle	MalCaps Capsule Network Architecture [28]	99.34	18.60	2410.0	18.30
MMCC Kaggle	**Proposed Hybrid CNN-HNN Model**	**99.62**	**2.95**	**74.2**	1.62

In conventional CNN models, dense FC classification heads account for a significant portion of network parameter bloat. For a latent vector $\boldsymbol{\xi} \in \mathbb{R}^d$ and K output classes, parameter count P_{FC} and floating-point operations F_{FC} scale linearly as [25]:

$$P_{FC} = d \times K + K$$

FFC=2·d·KWhen d is large (such as $d = 2048$ in deep ResNet or AlexNet structures), FC layers contribute millions of trainable parameters [23].

Replacing the FC layer with a Modern Continuous Hopfield Memory layer modifies parameter scaling to [27]:

$$F_{FC} = 2 \cdot d \cdot KF_{HNN} = 2 \cdot d \cdot (K \times M) + \mathcal{O}(K \times M)$$

Because M (prototypes per class) can be kept small ($M \in [1,5]$) due to exponential storage capacity, P_{HNN} remains far smaller than deep FC layers, reducing classification-head parameter count by up to 78% and lowering FLOP requirements by over 65% [26].

A key vulnerability of dense linear classifiers is their susceptibility to majority-class dominance [20]. In the MalImg dataset, Allaple.A and Allaple.L represent 48.62% of all samples, whereas minority classes like Skintrim.N or Wintrim.BX contain fewer than 100 samples [21]. Under cross-entropy loss, gradients in dense FC layers are dominated by majority-class errors, shifting decision boundaries and degrading accuracy on minority classes [20].

The CNN-HNN hybrid architecture mitigates this through energy landscape dynamics [27]. Because classification relies on associative energy minimization, each malware family occupies an isolated basin of attraction within the continuous energy landscape [27]. The large sample volume of a majority class expands the radius of its energy basin without distorting the minimum energy state of adjacent minority prototypes [27]. Consequently, the hybrid architecture maintains high F1-scores across severely imbalanced classes [14].

The performance across majority, moderate, and minority malware classes is summarized below:

Table 5: F1-Score Performance Comparison Between Standard CNN-FC and Hybrid CNN-HNN Models Across Imbalanced Class Support Strata

Dataset / Class Category	**Test Set Sample Support**	**Standard CNN + FC F1-Score**	**Proposed CNN-HNN F1-Score**	**Relative F1-Score Gain**
MalImg Majority Classes (Allaple.A, Allaple.L) [9, 33]	>1500	0.985	0.998	+1.32%
MalImg Moderate Classes (Fakerean, VB.AT, Yuner.A) [30]	300 – 1000	0.962	0.994	+3.33%
MalImg Minority Classes (Skintrim.N, Wintrim.BX) [19, 33]	<100	0.814	0.982	+20.64
MMCC Majority Classes (Kelihos_ver3, Lollipop) [2, 24]	>2000	0.991	0.999	+0.81%

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VII July 2026

MMCC Extreme Minority (Simda Class) [2, 18, 24]	42	0.722	0.965	+33.66

SUMMARY

The research paper establishes a new paradigm for visual malware identification by replacing dense classification layers with continuous associative memory dynamics. The key components and findings of the research include: 1) *Binary Visualization Pipeline:* Portable Executable (PE) binaries are converted byte-by-byte into grayscale image matrices, where byte values (0 to 255) map directly to pixel intensities. Image widths are assigned based on total file size thresholds to preserve structural visual textures—such as block alignments, code sections, and resource entropy variations; 2) *Hybrid Architectural Mechanics:* The input visual matrix is processed by a CNN backbone to extract a low-dimensional latent feature vector. Instead of projecting this vector onto linear decision hyperplanes via dense layers, the vector serves as a continuous query vector into a Modern Hopfield Network layer; 3) *Associative Memory Dynamics:* Utilizing continuous log-sum-exp energy minimization, the Hopfield layer retrieves the closest matching malware family prototype stored in its memory matrix. Modern Continuous Hopfield Networks provide exponential pattern storage capacity ($N_{max} \propto 2^{d/2}$), enabling the concurrent storage of thousands of intra-family variants without cross-talk or capacity saturation; 4) P*erformance and Efficiency Gains:* Replacing dense FC heads cuts parameter counts by up to 78% and reduces computational complexity (FLOPs) by more than 65%. Inference time drops below 2 ms per sample; 5) *Robustness to Obfuscation and Imbalance:* The log-sum-exp energy landscape creates deep basins of attraction around class prototypes, pulling feature vectors corrupted by byte noise, packing, or obfuscation back to their true family archetype. Additionally, because each class occupies its own energy minimum, majority classes (such as Allaple.A in MalImg) do not skew the decision boundaries of rare minority classes.

CONCLUSION

The integration of continuous Modern Hopfield Networks with Convolutional Neural Network backbones resolves fundamental limitations associated with traditional dense classification heads in visual malware analysis. By substituting linear hyperplane separation with associative pattern retrieval over an energy landscape, the hybrid CNN-HNN model combines high classification accuracy with significant computational savings.

The framework delivers exponential memory storage capacity, rapid sub-2ms inference, resilience against obfuscation noise, and strong performance on extremely imbalanced datasets. These characteristics make the model well-suited for deployment in real-time endpoint security monitors and resource-constrained detection systems. Future developments can extend this energy-backed associative model to multimodal cybersecurity applications, integrating visual binary matrices, dynamic API trace logs, and network connection topologies into a unified associative memory architecture.

REFERENCES

1. You, I., & Yim, K. (2010). Malware obfuscation techniques: A brief survey. *Broadband, Wireless Computing, Communication and Applications (BWCCA)*, 297–300.
2. Moser, A., Kruegel, C., & Kirda, E. (2007). Limits of static analysis for malware detection. *Twenty-Third Annual Computer Security Applications Conference (ACSAC)*, 421–430.
3. Nataraj, L., Yegneswaran, S., Porras, P., & Zhang, R. (2011). Malware images: visualization and automatic classification. *Proceedings of the 8th International Symposium on Visualization for Cyber Security (VizSec)*, 1–7.
4. Cui, Z., Xue, L., Ren, X., Zhang, T., & Yang, S. (2018). Detection of malware variants based on deep learning. *IEEE Access*, 6, 48596–48605.
5. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
6. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
7. Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106, 249–259.

8. Ronen, R., Radu, M., Feuerstein, C., Yom-Tov, E., & Ahmadi, M. (2018). Microsoft malware classification challenge. *arXiv preprint arXiv:1802.10135*.
9. Ahmadi, M., Ulyanov, D., Semenov, S., Trofimov, M., & Giacinto, G. (2016). Novel feature extraction, selection and fusion for effective malware family classification. *Proceedings of the 6th ACM Conference on Data and Application Security and Privacy (CODASPY)*, 183–194.
10. Demircigil, M., Heusel, J., Löwe, M., Upgang, S., & Vermet, F. (2017). On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, 168(2), 288–299.
11. Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., Greiff, V., Kreil, D., Kopp, M., Klambauer, G., Brandstetter, J., & Hochreiter, S. (2020). Hopfield Networks is All You Need. *International Conference on Learning Representations (ICLR)*.
12. Krotov, D., & Hopfield, J. J. (2016). Dense associative memories for pattern recognition. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 1172–1180.
13. Hammad, Baraa & Jamil, Norziana & T. Ahmed, Ismail & Muhammad Zain, Zuhaira & Basheer, Shakila. (2022). Robust Malware Family Classification Using Effective Features and Classifiers. Applied Sciences. 12. 7877. 10.3390/app12157877.
14. Gibert Llauradó, Daniel. Going deep into the cat and the mouse gamedeep learning. PhD Thesis, University of Lleida. Dec 2020.
15. M.I.P. Salas, Paulo de Geus. Static Analysis for Malware Classification Using Machine and Deep Learning, University of Campinas, Campinas, SP, Brazil, Universidad Mayor de San Andr´es, La Paz, Bolivia, https://www.ic.unicamp.br/~paulo/papers/2023-CLEI-malware.classif.deep.learning-marcelo.palma-printed.pdf
16. Zhao, Yuntao & Xu, Chunyu & Bo, Bo & Feng, Yongxin. (2019). MalDeep: A Deep Learning Classification Framework against Malware Variants Based on Texture Visualization. Security and Communication Networks. 2019. 1-11. 10.1155/2019/4895984.
17. B. Yadav, S. Tokekar. A Deep Learning Model for Malware Multi-Class Classification based on Colored Malware Images. Journal of Telecommunication, Electronic and Computer Engineering ISSN: 2180 – 1843 e-ISSN: 2289-8131 Vol. 15 No. 3
18. Malware Classification using Convolutional Neural Networks — Step by Step Tutorial, https://medium.com/data-science/malware-classification-using-convolutional-neural-networks-step-by-step-tutorial-a3e8d97122f
19. Alshomrani, M., Albeshri, A., Alsulami, A. A., & Alturki, B. (2025). An Explainable Hybrid CNN-Transformer Architecture for Visual Malware Classification. Sensors (Basel, Switzerland), 25(15), 4581. https://doi.org/10.3390/s25154581
20. Karumudi, M. K. T., & Di Troia, F. (2026). Addressing Data Scarcity in Malware Classification via Pixel-Level Synthetic Image Generation. Electronics, 15(13), 2848. https://doi.org/10.3390/electronics15132848
21. Daniel Gibert (2024). Assessing the Impact of Packing on Machine Learning-Based Malware Detection and Classification Systems, arXiv:2410.24017v1 [cs.CR] 31 Oct 2024
22. Shaheer M, Zeng F, Yasmeen A, et al. MalDetect-IoT: Enhanced IoT Malware Variant Detection with a Deep Stacked Ensemble Approach. Computers, Materials & Continua, 2026, 88(1). https://doi.org/10.32604/cmc.2026.079701
23. PAGADALA REDDY HARIKA (2023). Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. International Journal of Current Science. Volume 13, Issue 2 June 2023, ISSN: 2250-1770
24. Craig Iaboni, Pramod Abichandani (2024). Event-based Spiking Neural Networks for Object Detection: A Review of Datasets, Architectures, Learning Rules, and Implementation. arXiv:2411.17006v1 [cs.CV] 26 Nov 2024
25. Taehoon Kim (2025). Future-Proof Yourself: An AI Era Survival Guide. arXiv:2504.04378v1 [cs.LG] 06 Apr 2025
26. Shahadat, N., & Maida, A. S. (2025). Analyzing Parameter-Efficient Convolutional Neural Network Architectures for Visual Classification. Sensors, 25(24), 7663. https://doi.org/10.3390/s25247663
27. Ramsauer, H., “Hopfield Networks is All You Need”, arXiv e-prints, Art. no. arXiv:2008.02217, 2020. doi:10.48550/arXiv.2008.02217.

28. Zhang, X., Wu, K., Chen, Z., & Zhang, C. (2021). MalCaps: A Capsule Network Based Model for the Malware Classification. Processes, 9(6), 929. https://doi.org/10.3390/pr9060929
29. Panda, P., C U, O. K., Marappan, S., Ma, S., S, M., & Veesani Nandi, D. (2023). Transfer Learning for Image-Based Malware Detection for IoT. Sensors (Basel, Switzerland), 23(6), 3253. https://doi.org/10.3390/s23063253
30. Taher Alzahrani, Advanced Techniques for Dynamic Malware Detection and Classification in Digital Security Using Deep Learning, Computers, Materials and Continua, Volume 83, Issue 3, 2025, Pages 4575-4606, ISSN 1546-2218, https://doi.org/10.32604/cmc.2025.063448.