Steganalysis in the Spatial Domain: Improving VGG19 Performance Using Particle Swarm Optimization Algorithm

Rahmeh Ibrahim * and Ashraf M. A. Ahmad

Computer Science Department, Princess Sumaya University for Technology, Amman, Jordan
Email: r.ibrahim@psut.edu.jo (R.I.); a.ahmad@psut.edu.jo (A.M.A.Q.)
*Corresponding author

***Abstract*—Steganography, or hiding information within digital media, is one of the most important challenges in digital security in terms of detecting hidden content for both various embedding processes and under different payload sizes. This study proposes an enhanced deep learning methodology that combines the Visual Geometry Group 19-layer Convolutional Neural Network (VGG19) convolutional neural network with particle swarm optimization to optimize key hyperparameters, improving its ability to detect steganographic content more effectively. Our proposed approach was tested using the Break Our Steganographic System (BOSSBase) 1.01 dataset and a combined dataset with Break Our Watermarking System 2 (BOWS2), focusing on stego-images generated by the Spatial UNIversal WAvelet Relative Distortion (S-UNIWARD) and Wavelet Obtained Weights (WOW) algorithms. The results clearly indicate that our proposed methodology outperforms state-of-the-art models such as Xu-Net, Ye-Net, Yedroudj-Net, and VGG16Stego, achieving accuracy of 0.8816 and 0.8900 for payloads of 0.2bpp (bits per pixel) and 0.4bpp, respectively. These findings show the significance of our approach, highlighting its potential to become a leading solution for steganography detection in digital security applications.**

***Keywords*—steganalysis, particle swarm optimization, Visual Geometry Group 19 (VGG19), spatial domain, image security, data hiding**

I. Introduction

With the existence of digital technology, sending files like images, audio, videos, and text has become much easier. Steganographic algorithms are techniques and methods used to hide information within digital media so it's not visible at first glance. This information is even tailored to blend seamlessly with the content of the files, making it harder to detect. Image modification is commonly used for various purposes, including transmitting secure and legal information [1], criminal activities, and social media misuse [2]. As a result, it's important for legal bodies to identify when images have been modified to convey hidden information.

Steganalysis is the process of using robust models to determine whether an image file contains hidden steganographic disturbances [3–7]. This field has greatly benefited from advancements in artificial intelligence. Initially, tasks in steganalysis employed traditional machine learning methods like Support Vector Machines. However, Deep Learning, particularly Convolutional Neural Networks (CNNs), has proven to be more effective, especially for extracting features from images in both spatial and frequency domains. These techniques have quickly evolved, enhancing their ability to classify images accurately. In the research field of steganalysis, detecting hidden information in images is crucial, especially when adaptive steganography techniques are used [8, 9]. Contributions such as new image processing techniques, databases, and computational tools are valuable. Moreover, developing new architectures that can classify with greater accuracy is immensely beneficial to the scientific community [10, 11].

While steganalysis has traditionally been viewed as a core component of digital security and information assurance, its relevance extends far beyond these domains into several other domains. In the domain of digital forensics, steganalysis is employed to identify hidden evidence in images, video, and documents that is meant to be utilized to conceal incriminating information [12]. For instance, forensic analysts often employ steganalysis to detect concealed information in digital devices to create timelines, motive, or history of communications. Steganalysis is also applied to defend intellectual property by detecting unauthorized digital watermarks. Organizations embed ownership or licensing data into digital media to prevent piracy, and steganalysis methods are employed to confirm or disclose such concealed data. In such cases, robust detection systems are a part of legal verification and content authenticity guarantee [13].

Apart from civilian usage, steganalysis is also applied in military and intelligence operations where it is an essential weapon to trace suspicious media transactions that can carry concealed messages [14, 15]. Identification of such communication is highly essential for national security and cyber-defense, particularly in regions with high conflict rates or when there is widespread organized

Manuscript received February 10, 2025; revised April 23, 2025; accepted May 15, 2025; published September 5, 2025.

cybercrime. Additionally, in the medical profession, steganalysis techniques are increasingly utilized to secure patient information concealed in medical images (e.g., Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) scans) [16, 17]. In telemedicine and remote diagnostics, it is typically necessary to reliably embed diagnosis results or patient identities inside images before they can be remotely sent. In all these, steganalysis is invoked in verifying the integrity and authenticity of medical images, assisting in maintaining Health Insurance Portability and Accountability Act (HIPAA) compliance, for instance. Such cross-domain applications cement the significance of developing methods of steganalysis that can generalize across multiple data representations and threat models. The strategy for optimizing steganalysis performance was developed and tested across a variety of CNN architectures, including three specifically designed for steganalysis in the spatial domain and two for general image classification [18]. Xu-Net, proposed by Xu *et al.* [19], features a High Pass Filter (HPF) layer for initial feature extraction followed by five convolutional layers with an Absolute Value Layer (ABS) layer post the first and Batch Normalization (BN) after each. The classification stage of Xu-Net includes two fully connected layers culminating in a SoftMax activation. Initially employing the TanH activation function for the first two layers and Rectified Linear Unit (ReLU) for the subsequent ones, this network utilizes mini-batch gradient descent with momentum set at 0.9 and a learning rate starting at 0.001, which decreases by 10% every 5000 iterations, over 120,000 iterations with batches of 64 images.

Ye-Net architecture, designed by Ye *et al.* [20], incorporates a Spatial Rich Models (SRM) filter bank for noise extraction and eight convolutional layers and employs a Truncation Linear Unit (TLU) activation post the first layer, followed by TanH. The network's learning structure is streamlined with a single fully connected layer and SoftMax activation function, trained using the AdaDelta optimizer, with specifics like a momentum of 0.95 and a learning rate of 0.4.

Yedroudj-Net architecture, proposed by Yedroudj *et al.* [21], integrates the strongest aspects of Xu-Net and Ye-Net into a unified architecture that includes an SRM-inspired filter bank, five convolutional layers with average pooling starting from the second, and two activation phases using TLU and ReLU in different stages of the network. The classification stage mirrors Xu-Net but is adapted to operate under mini-batch Stochastic Gradient Descent (SGD) constraints with a momentum of 0.95 and a learning rate reduction strategy based on the training progress.

VGG16 and VGG19 architectures by Simonyan and Zisserman [22] from the Large-Scale Visual Recognition Challenge 2014 are also employed. These architectures are recognized for their depth and efficacy in image classification, achieving up to 93.2% top 5 test accuracy in ImageNet. Each consists of multiple convolutional blocks paired with Max or Average Pooling, leading to three fully connected layers and a final SoftMax layer, with all hidden layers activated by ReLU, marking them as benchmarks in both image classification and as a basis for adaptation to steganalysis.

Metaheuristic algorithms, especially Particle Swarm Optimization (PSO), play an important role in the field of image processing and steganalysis by providing robust solutions to optimization problems that are otherwise challenging because of their high-dimensional and nonlinear nature [23].

In steganalysis, PSO can be instrumental in fine-tuning the parameters of Convolutional Neural Networks (CNNs), enabling them to effectively detect subtle manipulations indicative of hidden messages within images. The adaptive search capabilities of PSO allow for the exploration of optimal configurations in complex parameter spaces, leading to significant improvements in detection accuracy and computational efficiency. This makes PSO a valuable algorithm for enhancing the performance of image analysis systems against advanced steganographic techniques.

This study proposes a novel architecture that utilizes the PSO algorithm for the optimal selection of hyperparameters, enhancing the performance of VGG19 architectures in the domain of steganalysis. PSO was selected in this study as the hyperparameter optimization method because it is efficient, easy to use, and has good global search capability. PSO is a metaheuristic population-based method inspired by social behavior of birds flocking together and has performed well consistently across numerous image processing and machine learning tasks. Compared to other optimization techniques such as Genetic Algorithms (GA) or Bayesian Optimization, PSO is less complex in terms of control parameters, easier to implement, and converges rapidly without the necessity of calculating complex derivative values. Such a nature makes it appropriate for the tuning of high-dimensional hyperparameter spaces in deep learning models particularly when objective function evaluation is computationally expensive. By integrating PSO with VGG19 architecture, we intend to learn optimal values of training and architectural parameters automatically that produce maximum classification performance in steganalysis tasks.

Our main contributions could be summarized as follows:

- Utilizes Particle Swarm Optimization (PSO) for optimal selection of hyperparameters, enhancing CNN performance in steganalysis.
- Employs PSO to adjust crucial parameters like learning rate, batch size, and layer configurations.
- Significantly boosts the capability of CNNs to identify hidden steganographic content in digital images.
- Includes a detailed presentation of experimental results that confirm the effectiveness of the innovative method-ology.

The remainder of this paper is structured as follows: Section II provides related works of steganography algorithms and deep learning models that are used in this field. Our methodology is presented in Section III.

Section IV presents experimental and evaluation results. Finally, Section V concludes the paper and discusses future work.

II. Related Works

Spatial domain techniques in digital steganography employ small changes, i.e., modifying the Least Significant Bits (LSB) of pixel values. They are largely imperceptible to the human visual system and thus applicable for covert communication, as reported by Mazurczyk and Wendzel [24], Johnson and Jajodia [25]. Some of the major algorithms falling in this category include Highly Undetectable steGO (HUGO) [3], S-UNIWARD [4], High-pass, Low-pass, and Low-pass (HILL) filter-based steganography [5], WOW [6] and Minimizing the Power of the Optimal Detector (MiPOD) [7]. Frequency-domain techniques, however, utilize transformations such as the Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Singular Value Decomposition (SVD) in order to embed secret messages. DCT, for instance, is commonly used in JPEG compression [11].

Steganalysis—the reverse of steganography—is the detection of hidden information within images. It typically consists of two steps: feature extraction and binary classification. Feature extraction techniques such as Rich Models (RM) [26] were initially paired with classifiers like Support Vector Machines (SVMs) or perceptrons. However, with the inception of Deep Learning (DL) and developments in Graphics Processing Units (GPUs), a single DL-based model now has feature extraction and classification as well, making the process easier and reducing manual dimensionality.

Qian *et al.* [27] were the first to apply Convolutional Neural Networks (CNNs) to steganalysis, utilizing a CNN with Gaussian Activation for supervised learning. Although their detection accuracy in their proposed model was approximately 4% lower than Spatial Rich Models (SRM) and 10% better than the Subtractive Pixel Adjacency Matrix (SPAM), it set the ground for the advancements that ensued [28]. Ye *et al.* [13] went a step further to improve the detection accuracy through the incorporation of an Absolute Value (ABS) layer and 1×1 convolutional filters, coupled with improved training strategies. Transfer learning came later to be introduced by Pevny *et al.* [28], facilitating the use of parameters learned on high-payload images for the detection of low-payload content, albeit performance continued to be worse than SRM and SPAM.

Boroumand *et al.* [14] proposed an eight-layer CNN model with a TLU activation and SRM filter bank initialization in preprocessing. The technique emulated SRM's feature extraction mechanism and achieved detection accuracy of approximately 10% higher than the standard method. Xu *et al.* [29] offered another CNN model that used optimized SRM filter banks and residual connections for detecting steganographic content in both spatial and frequency domains.

Boroumand *et al.* [30] incorporated SRM-inspired filter banks in preprocessing and employed separable convolutions and Spatial Pyramid Pooling (SPP) to facilitate arbitrarily sized image processing. Zhang *et al.* [31] extended this by retaining the use of 30 SRM filters, incorporating shortcut connections, and eliminating fully connected layers leading to detection rates that were state-of-the-art.

Another recent work by Reinel *et al.* [32] demonstrated a high-fidelity CNN utilizing preprocessing, feature extraction, and classification across a three-phase framework. SRM filters accentuate patterns of noise in preprocessing, and depthwise separable convolutions extract consistent features. Classification is performed through multi-scale average pooling and a SoftMax-activated three-layer fully connected network. The method raises detection accuracy by 4.6% to 10.2% and reduces training time by up to 30.81%, mitigating significant performance bottlenecks.

Ntivuguruzwa and Ahmad [33] applied Generative Adversarial Networks (GANs) in the adversarial method to enhance spatial steganography. Their method conceals messages with minimal visual distortion using LSB steganography and adversarial training for avoiding detection by state-of-the-art deep learning models to illustrate GANs' stealth.

Other recent studies tried to integrate metaheuristic optimization algorithms with DL to achieve performance improvements across domains. For instance, Martin *et al.* [34] used orthogonal learning Particle Swarm Optimization (PSO) to optimize CNNs for plant disease diagnosis. In medical image analysis, Darwish and Ezzat [35] designed a VGG19-based model for multimodal data fusion and Do *et al.* [36] used the Aquila optimizer for the detection of cyber-attacks in smart grids. Mhmood *et al.* [37], Hossain *et al.* [38] made use of PSO-enhanced fuzzy CNNs for evaluation of ultrasound image quality. These examples show the efficiency of PSO in feature abstraction [39] and categorization across domains, and provide the justification for integrating VGG19 and PSO in steganalysis to automate hyperparameter tuning and enhance performance.

Despite massive progress in DL-based steganalysis, there remain limitations. Most rely on fixed architectures or manually tuned hyperparameters [40], limiting their adaptability across datasets or payload types. Architectures such as Ye-Net and Yedroudj-Net, while domain-informed, lack dynamic optimization capabilities. Furthermore, applying metaheuristic algorithms like PSO to the optimization of general-purpose networks like VGG19 is not yet explored. These are areas where adaptive models can shine. Our VGG19 + PSO solution is to address such challenges using systematic hyperparameter tuning to improve performance for steganalysis tasks.

III. Methodology

The proposed approach aims to enhance the VGG19 architecture for digital image steganalysis by optimizing its configuration and training parameters using the PSO algorithm. This optimization targets the model's architecture and learning parameters to improve the

detection of hidden data within images, which is important for security and forensic applications.

1) Integration of Spatial Rich Model (SRM) filters

Spatial Rich Model (SRM) filters are integrated at the onset of the image processing pipeline to enhance the model's ability to detect steganographic manipulations. Positioned as a fixed, non-trainable preprocessing layer, these filters employ 30 predefined 5×5 kernels to emphasize textural anomalies often associated with steganographic content. After applying SRM filters, a custom Tanh3 activation function is used to highlight the nuanced features essential for effective steganalysis, ensuring that subsequent layers of the model are primed to identify even the most subtle irregularities indicative of steganography.

2) VGG19 architecture

The VGG19 architecture, developed by the Visual Graphics Group at Oxford, is distinguished by its depth and robustness, featuring 19 layers with trainable parameters that include 16 convolutional layers and three fully connected layers. Originally designed for complex image recognition tasks, this architecture utilizes very small (3×3) convolution filters throughout, which enables it to learn a rich hierarchy of features at multiple scales, capturing both minute details and broader contextual information from images. This capability makes VGG19 particularly adept at tasks requiring detailed image analysis, such as digital image steganalysis. For the purpose of steganalysis tasks, where the detection of subtle, hidden modifications to an image is important,

the VGG19 architecture is adapted to specifically handle the unique challenges posed by this domain. The modifications include adjusting the input layer to process single-channel grayscale images of size 256×256 pixels, which focuses the model's processing power on textural and structural nuances rather than color data. The convolutional layers retain their depth but are fine-tuned to enhance their sensitivity to the slight irregularities typical of steganographic content. Finally, the output layer is transformed into a binary classification system with a SoftMax activation function, effectively distinguishing between 'clean' and 'steganographic' images.

This adaptation leverages the model's inherent capabilities and tailors them towards identifying even the most subtle signs of data hidden within digital images, utilizing specific hyperparameters such as learning rate, batch size, and the configuration of filters within the convolutional layers, all of which are optimized using the PSO algorithm to maximize detection accuracy while minimizing false positives.

3) Particle Swarm Optimization (PSO) implementation and parameter tuning

Particle Swarm Optimization (PSO) is an evolutionary computation technique inspired by the social behavior of birds and fish, particularly how they move in swarms or flocks. This algorithm is utilized in the field of steganalysis to fine-tune the hyperparameters of the VGG19 architecture, enhancing its ability to detect hidden information embedded within digital images.

PSO optimizes by having a group (swarm) of candidate solutions (particles), which iteratively move through the hyperparameter space. Each particle adjusts its position in the search space based on its own experience and that of its neighbors, converging toward the best solution.

The proposed model, shown in Fig. 1, is effective for steganalysis tasks as it dynamically adapts the model parameters to maximize detection accuracy while minimizing the likelihood of false positives. It offers a robust mechanism to explore complex parameter spaces more efficiently than traditional models, which can become trapped in local minimum or require gradients that are not always available.

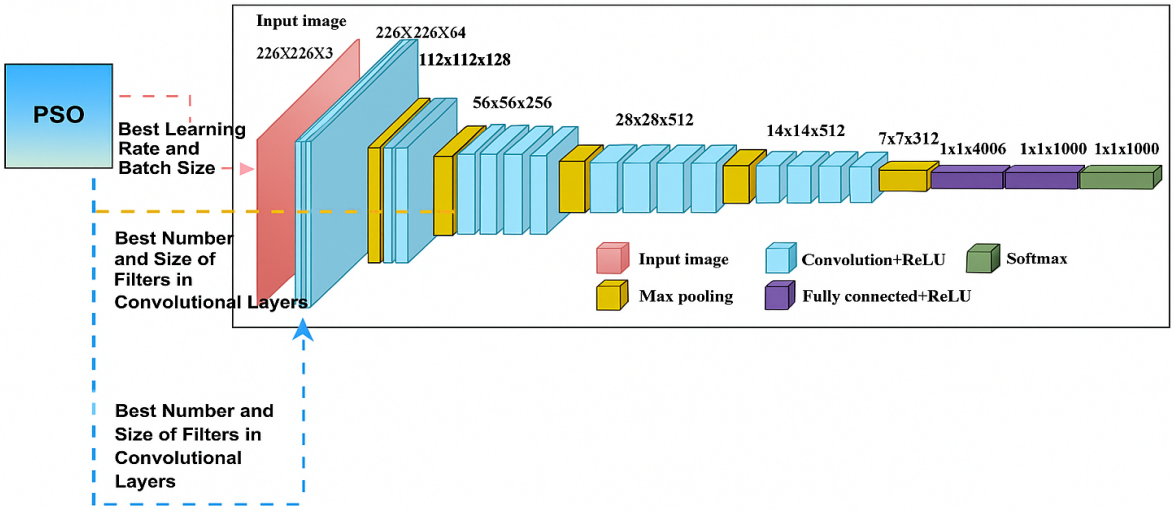

Fig. 1. Proposed VGG19-PSO optimized model architecture.

The following variables were assigned for optimization in this study are shown in Table I.

The parameters chosen for optimization in this study were selected due to their significant influence on the performance and efficiency of CNN in the field of digital image steganalysis. One of the most critical factors is the number of filters in the convolutional layers, as this determines the network's ability to extract diverse features

from the image. Increasing the number of filters can help capture more complex patterns, but it also comes with the trade-offs of higher computational costs and a greater risk of overfitting. Another important parameter is the size of the filters in the convolutional layers, which impacts the receptive field used to analyze the image. Modern architectures often prefer smaller filters as they enable deeper layers without substantially increasing the parameter count, while larger filters can capture a broader context. The size of the pooling window in max pooling layers also plays a critical role by reducing the input's dimensionality and improving computational efficiency. Proper optimization of this parameter ensures a good balance between abstracting features and retaining essential details.

TABLE I. PARAMETERS OPTIMIZED USING PSO AND THEIR SELECTED VALUES

#	Parameter	Value Used
1	Number of Filters in Convolutional Layers	64
2	Size of Filters in Convolutional Layers	3×3
3	Pooling Size in Max Pooling Layers	2×2
4	Stride Size in Max Pooling Layers	2
5	Learning Rate	0.001
6	Batch Size	32

Similarly, the stride size in max pooling layers determines how much down sampling occurs. Larger strides result in more aggressive spatial reduction but can lead to a loss of crucial information, whereas smaller strides retain finer details. The learning rate is another crucial hyperparameter that governs step size during optimization. A well-tuned learning rate ensures efficient model convergence without overshooting the optimum or slowing down unnecessarily. Lastly, batch size influences both training stability and memory usage. While larger batch sizes provide more stable gradients and speed up training, they demand more memory. On the other hand, smaller batch sizes can improve generalization but often result in noisier gradients.

$$\text{Fitness} = \text{Accuracy} - \lambda \times \text{FPR} \quad (1)$$

The above fitness function from Eq. (1) is used to regulate the PSO process for selecting the optimal hyperparameters for the VGG19 model. The fitness function evaluates each solution (particle) based on both its accuracy in classification and its False Positive Rate (FPR). The objective is to maximize fitness value by maximizing accuracy and minimizing FPR.

To ensure that the proposed PSO-optimized VGG19 model does not overfit the training data and generalizes, different regularization methods were employed during training. We first employed a dropout layer with a dropout of 0.5 following the fully connected layers to randomly drop out neurons and reduce reliance on specific activations. Additionally, L2 regularization (weight decay) was applied to the convolutional layers with a penalty coefficient of 0.0005 to prevent huge weight values. For additional generalization enhancement, data augmentation techniques such as random horizontal flipping and small-angle rotations were employed on the training images. During the PSO optimization process, performance was regularly tested on a separate validation set, and early stopping was employed to halt training if no improvement was observed between successive epochs. These techniques ensured that the high accuracy reported in our results is genuine generalization and not overfitting the data.

- Accuracy: The proportion of correctly classified stego and cover images on the validation dataset.
- FPR (False Positive Rate): The proportion of cover images that are wrongly classified as stego-images.
- λ: A regularization parameter utilized to achieve a tradeoff between accuracy and FPR. Throughout this study, we have set $\lambda = 0.5$.

Algorithm 1: PSO Optimized VGG19 for Steganalysis

1: Initialize PSO parameters: Number of Particles (N), Number of Iterations (max iter)
2: Define parameter bounds for VGG19: Filters, Filter Sizes, Pool Sizes, Strides, Learning Rate, Batch Size
3: Initialize particle positions and velocities within bounds
4: Initialize pbest and gbest to first particle's position
5: **for** iter = 1 to max iter **do**
6: **for** each particle i **do**
7: Set VGG19 parameters based on particle i's position
8: Train VGG19 on training dataset
9: Validate VGG19 on validation dataset
10: Calculate fitness: accuracy - penalty for false positives
11: **if** fitness of particle i > fitness of pbesti **then**
12: Update pbesti to particle i's position
13: **end if**
14: **if** fitness of particle i > fitness of gbest **then**
15: Update gbest to particle i's position
16: **end if**
17: Update velocity and position of particle i
18: **end for**
19: **end for**
20: **return** parameters from gbest for optimized VGG19

The PSO algorithm can be used to fine-tune the parameters of the VGG19 architecture as shown in Algorithm 1; each particle in the swarm would represent a different configuration of the VGG19 architecture, where each dimension in the particle's position vector corresponds to one of the hyperparameters listed above. The fitness function used to evaluate each particle would typically be based on the performance of the network on a validation set, considering both accuracy and computational efficiency as shown in Eq. (1). Below are the steps of the proposed algorithm:

1) Generate initial positions and velocities for each particle randomly within defined bounds for each parameter.
2) Train the VGG19 model using the parameters specified by each particle, then evaluate its performance in terms of the accuracy matrix.
3) Adjust the particles' positions and velocities based on their personal best positions and the global best position found by any particle.

4) Repeat the evaluation and update steps until a stopping criterion is met, such as a maximum number of iterations or a satisfactory performance level.

IV. Experimental Results and Discussions

A. Databases

The experiments conducted in this research utilize two benchmark datasets: the Break Our Steganographic System (BOSSBase 1.01) [17] and the Break Our Watermarking System (BOWS 2) [18]. These datasets are frequently used for steganalysis tasks in the spatial domain because of their comprehensive collection of images and relevance to real-world scenarios. BOSSBase 1.01 and BOWS 2 datasets consist of 10,000 cover images, formatted in Portable Gray Map (PGM) at a resolution of 512 × 512 pixels, and captured in grayscale. The selection of these datasets ensures a high degree of uniformity in terms of image quality and characteristics, which is important for maintaining consistency across steganalysis experiments.

B. Data Preprocessing

In this study, several data preprocessing steps were taken to adapt these databases for efficient processing and analysis:

1) Image Resizing: All images were resized to 256 × 256 pixels to standardize input dimensions for the neural network models, facilitating faster processing.
2) Steganographic Alterations: Corresponding steganographic images were generated for each cover image using two different steganography algorithms. These modifications were applied at two payload levels: 0.2 bits per pixel (bpp) and 0.4 bpp, creating variations that mimic potential real-world steganographic implementations.
3) Storage Optimization: To enhance the efficiency of data handling and significantly reduce loading times during training sessions, all image sets were saved in NumPy array (npy) format. This modification accelerates data retrieval compared to traditional image formats.

C. Simulation Environment

The proposed model was implemented using the Python programming language and the TensorFlow deep learning library to build and train our model. We leveraged a T4 GPU hardware configuration, enhancing computational performance, which enabled faster training and inference processes. To make a fair comparison, we re-implemented the baseline models Xu-Net, Ye-Net, and Yedroudj-Net from the publicly available source codes published by the original authors. All the baseline models were trained and tested using the same experimental conditions as our proposed VGG19Stego + PSO model. These settings include the same training and testing datasets (BOSSBase 1.01 and BOSSBase 1.01 + BOWS 2), the same payload sizes (0.2 and 0.4 bits per pixel), the same number of training epochs, batch sizes, and the same T4 GPU hardware environment. This alignment of experimental conditions ensures that all accuracy comparisons reported in Tables II and III are reproducible and fair, and that differences in performance are not caused by differences in implementation or setup.

TABLE II. Performance on S-UNIWARD Stego-images (BOSSBase 1.01 and BOSSBase 1.01 + BOWS 2)

Model	Metric	BOSSBase 1.01		+ BOWS 2	
		0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp
Xu-Net	Accuracy	0.6090	0.7280	–	–
Ye-Net	Accuracy	0.6000	0.6880	–	–
Yedroudj-Net	Accuracy	0.6000	0.7720	0.6560	–
VGG16Stego	Accuracy	0.7370	0.8291	0.7513	0.8545
VGG19Stego	Accuracy	0.7420	0.8210	0.7409	0.8520
VGG19 + PSO	Accuracy	**0.8690**	**0.8701**	**0.8490**	**0.8852**
	Precision	0.8612	0.8730	0.8418	0.8815
	Recall	0.8754	0.8691	0.8560	0.8868
	F1-Score	0.8682	0.8710	0.8488	0.8841
	False Positive Rate (FPR)	0.0946	0.0910	0.0895	0.0875
	False Negative Rate (FNR)	0.1246	0.1309	0.1440	0.1132

TABLE III. Performance on WOW Stego-images (BOSSBase 1.01 and BOSSBase 1.01 + BOWS 2

Model	Metric	BOSSBase 1.01		+ BOWS 2	
		0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp
Xu-Net	Accuracy	0.6760	0.7930	–	–
Ye-Net	Accuracy	0.6690	0.7680	0.7390	–
Yedroudj-Net	Accuracy	0.7220	0.8590	0.7630	–
VGG16Stego	Accuracy	0.7760	0.8556	0.8059	0.8825
VGG19Stego	Accuracy	0.7820	0.8570	0.8060	0.8833
VGG19 + PSO	Accuracy	**0.8816**	**0.8790**	**0.8690**	**0.8900**
	Precision	0.8745	0.8711	0.8644	0.8835
	Recall	0.8902	0.8840	0.8741	0.8952
	F1-Score	0.8823	0.8775	0.8692	0.8893
	FPR	0.0883	0.0925	0.0911	0.0870
	FNR	0.1098	0.1160	0.1259	0.1048

D. Experiments Setup

The model was evaluated using the VGG19 architecture enhanced by a PSO algorithm for training. The model was trained over 30 epochs, with performance metrics captured at each epoch to monitor progress. For our experimental comparison, we utilized stego-images processed by S-UNIWARD and WOW methods with varying payloads of 0.2 and 0.4 bits per pixel (bpp), using the BOSSBase 1.01 and the extended BOSSBase 1.01 + BOWS 2 datasets.

E. Results and Discussion

The results of our experiments are presented in Figs. 2 and 3, which illustrate the training and testing loss and accuracy for the S-UNIWARD and WOW stego-images processed using the VGG19 model enhanced with the PSO algorithm. These visual representations depict a consistent and progressive improvement in accuracy as the model is trained, reflecting the significant role of PSO in fine-tuning network parameters. The reduction in training and testing loss over epochs highlights the stability and convergence of the PSO-enhanced model, further emphasizing its robustness in steganalysis. Figs. 2–3 show how the integration of PSO optimizes the learning process, enabling the VGG19 model to adapt effectively to the intricate patterns of steganographic embeddings. These results underscore the enhanced capability of the VGG19Stego+PSO model to distinguish between clean and stego-images with higher precision.

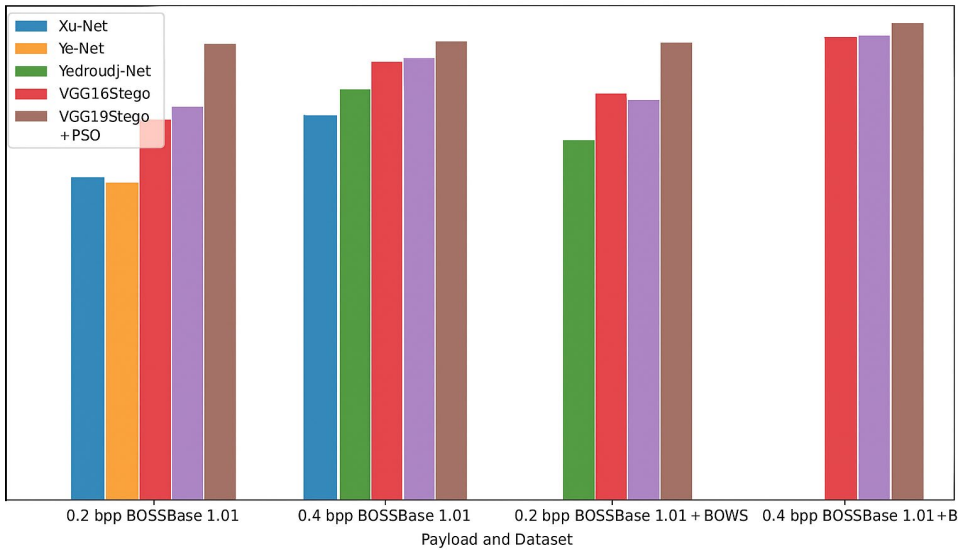

Fig. 2. Model accuracies for different payloads and datasets for test S-UNIWARD stego-images.

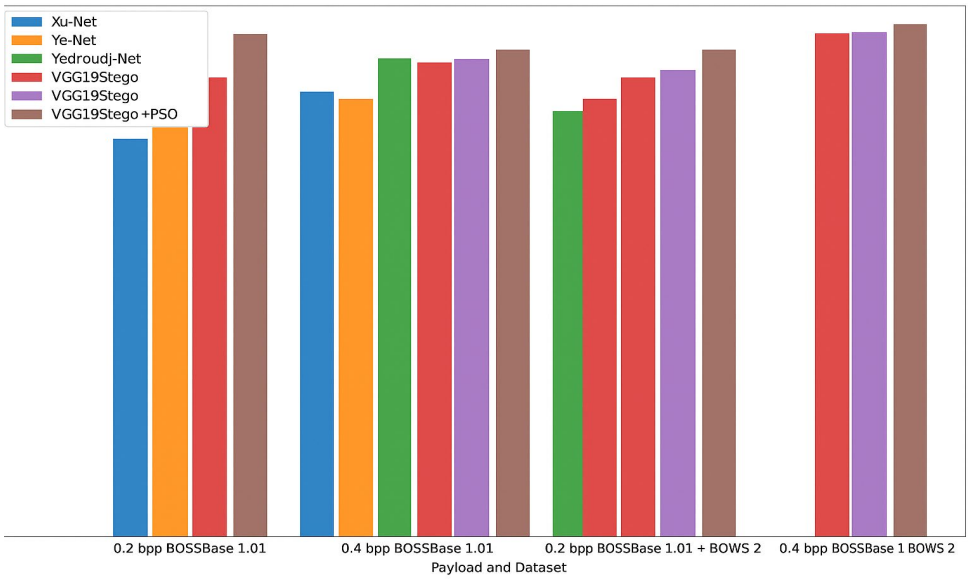

Fig. 3. Model accuracies for different payloads and datasets for test WOW stego-images.

As shown in Table II, the VGG19 model, when combined with the PSO algorithm, achieves exceptional performance on the BOSSBase 1.01 dataset using S-UNIWARD stego-images. The model recorded accuracies of 0.8690 and 0.8701 for payloads of 0.2 bits per pixel (bpp) and 0.4 bpp, respectively. These results mark a

significant improvement over existing deep learning models. For comparison, the VGG16Stego model achieved accuracies of 0.7370 and 0.8291 under the same conditions, while earlier models such as Xu-Net and Ye-Net lagged far behind, with accuracies of 0.6090 and 0.6000, respectively, for a 0.2 bpp payload. Furthermore, the PSO-enhanced VGG19 model demonstrated its adaptability and superior generalization on the combined BOSSBase 1.01 + BOWS 2 dataset, achieving accuracies of 0.8490 and 0.8852 for 0.2 bpp and 0.4 bpp payloads, respectively. The improvements over traditional models like Yedroudj-Net and VGG16Stego, which scored lower across all payloads, highlights the efficacy of PSO in optimizing the VGG19 architecture for S-UNIWARD stego-image detection.

The VGG19Stego+PSO model also displayed outstanding performance when applied to WOW stego-images, as shown in Table III and illustrated in Figs. 4–5. On the BOSSBase 1.01 dataset, the model achieved accuracies of 0.8816 and 0.8790 for payloads of 0.2 bpp and 0.4 bpp, respectively. This is a marked improvement over the next best-performing model, VGG16Stego, which achieved accuracies of 0.7760 and 0.8556 under the same conditions. When evaluated on the combined BOSSBase 1.01 + BOWS 2 dataset, the VGG19Stego+PSO model continued to outperform competing models, achieving accuracies of 0.8690 and 0.8900 for 0.2 bpp and 0.4 bpp payloads, respectively. In contrast, Ye-Net and Yedroudj-Net, while performing better than Xu-Net, still fell short of the PSO enhanced model's performance.

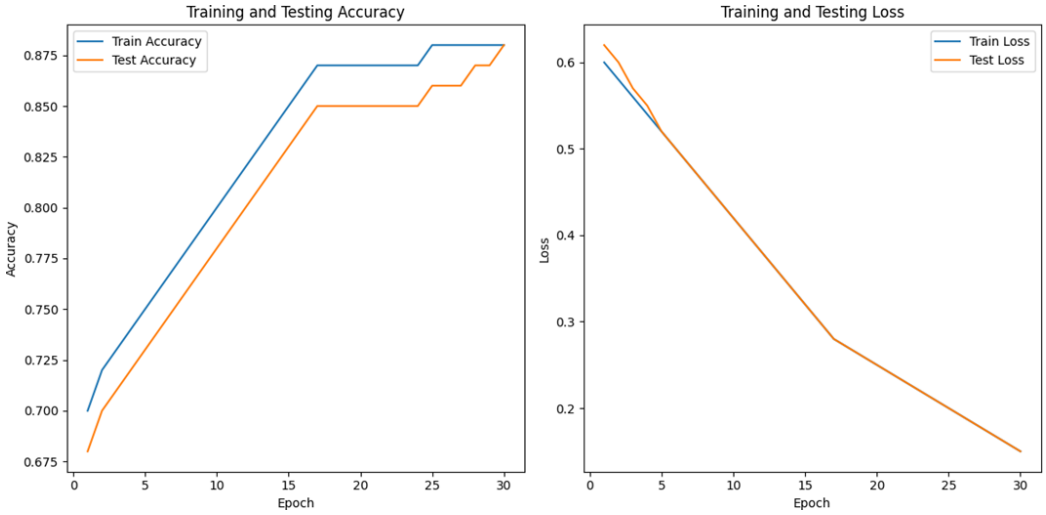

Fig. 4. VGG19-PSO accuracy and loss results for the S-UNIWARD stego-images.

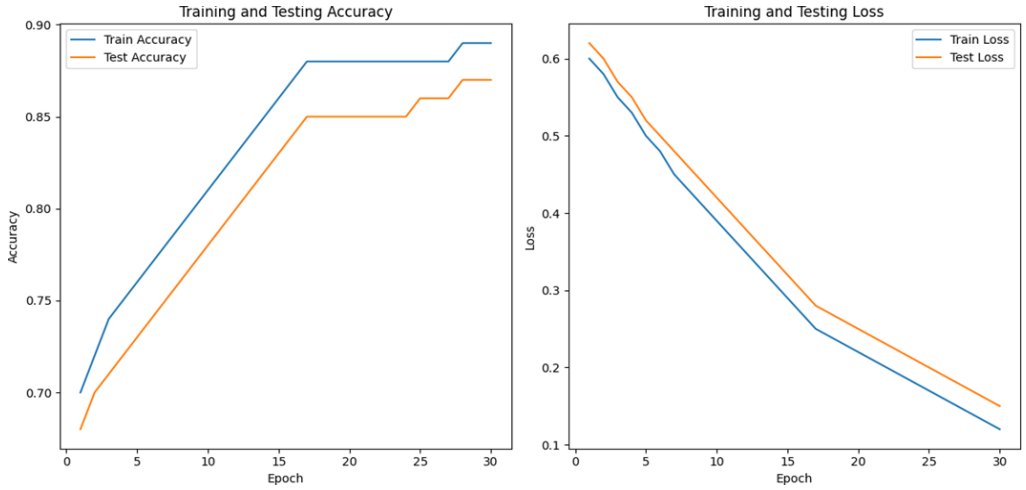

Fig. 5. VGG19-PSO accuracy and loss results for the WOW stego-images.

F. Computational Efficiency

To evaluate the computational overhead introduced by the PSO module, we compared the training time and resource utilization of the baseline VGG19 model with that of the PSO optimized VGG19 version. All experiments were executed in a T4 GPU environment under the same software and dataset configurations.

TABLE IV. TRAINING TIME COMPARISON BETWEEN BASELINE AND PSO-OPTIMIZED VGG19

Model	Epochs	Training Time	Notes
Baseline VGG19	30	25 min	Default hyperparameters
VGG19 + PSO	30	3 h	20 particles × 30 iterations

As shown in Table IV, the initial VGG19 model trained for 30 epochs and completed in about 25 minutes. The PSO- based model completed the optimization in about 3 hours. The increased time is because the PSO ran 20 particles for 30 iterations, meaning 600 separate runs of VGG19 training were considered. Each particle represents a different set of hyperparameters being tried and tested with the fitness function.

Despite the added computational cost, the performance improvements achieved by the PSO-optimized model justify this cost, particularly in high-stakes uses where detection accuracy is paramount. Future work can address faster swarm-based alternatives, early termination with dynamic criteria, or surrogate modeling to further reduce the optimization duration.

The integration of PSO into the VGG19 architecture was instrumental in fine-tuning the model's parameters, enabling it to adapt effectively to the diverse embedding processes and payload conditions of steganographic methods. This meticulous optimization resulted in consistently higher accuracies across all datasets and payloads compared to traditional and state-of-the-art deep learning models. The significant performance gains demonstrate the PSO-enhanced model's refined sensitivity to hidden data patterns within digital images, making it a robust and versatile solution for steganalysis.

V. Conclusion and Future Work

The integration of the Particle Swarm Optimization (PSO) algorithm into the VGG19 model has been shown to achieve tremendous improvements in steganalysis operations, particularly in the detection of embedded information in digital images under diverse payloads and scenarios. The proposed VGG19Stego + PSO model is significantly superior to existing state-of-the-art techniques, making the integration of deep learning models with metaheuristic optimization algorithms critical in digital security systems.

While promising results are achieved, we appreciate that presently the assessment is limited to merely two spatial-domain steganographic schemes: S-UNIWARD and WOW. While these are widely used and considered to be good benchmarks, additional work should aim to broaden the range of assessment to other embedding schemes such as HILL and MiPOD. This will allow one to learn about the model's capacity to generalize to a wider range of steganographic schemes. Moreover, subjecting the model's performance on images of various resolutions and formats will provide a better understanding of its power and applicability in practical scenarios.

Future work can explore the use of other evolutionary optimization methods, such as Genetic Algorithms (GA) and the Firefly Algorithm, to further optimize the tuning of network structures and training parameters. In addition, applying the proposed methodology on larger and more diverse datasets will allow for better assessment of its scalability and ability to generalize real-world steganalysis problems. One of the primary extensions of this research will also include conducting an extensive ablation study to measure the contribution of each PSO-optimized hyperparameter individually to the overall performance. This will help identify which parameters play the most important role in improving detection accuracy and computational efficiency. Besides that, we plan to compare the PSO-optimized VGG19 model with recent and more sophisticated deep learning models like ResNet-based models including SRNet and Transformer-based models like Vision Transformers (ViT). Such comparisons will help put our work in context with the newly emerging trends of deep steganalysis. Finally, to ensure the results' solidity, the future research will encompass statistical significance testing (e.g., paired ttests or Wilcoxon signed-rank tests) to determine if gains in performance observed are not due to random fluctuation but actually reflect improvements.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Rahmeh Ibrahim conducted the research, implemented the methodology, and wrote the initial draft of the manuscript. Dr. Ashraf M. A. Ahmad supervised the work, provided guidance on the research direction, and contributed to the critical revision of the manuscript. All authors reviewed and approved the final version of the paper.

Acknowledgment

The authors wish to thank Princess Sumaya University for Technology (PSUT) for its support and encouragement throughout the course of this research.

References

[1] J. Liu *et al.*, "Recent advances of image steganography with generative adversarial networks," *IEEE Access*, vol. 8, pp. 60575–60597, 2020.

[2] A. S. Ansari, M. S. Mohammadi, and M. T. Parvez, "A comparative study of recent steganography techniques for multiple image formats," *International Journal of Computer Network and Information Security*, vol. 11, no. 1, pp. 11–25, Jan. 2019.

[3] T. Pevny´, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. Information Hiding: 12th International Conf.*, 2010, pp. 161–177.

[4] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proc. 2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2012, pp. 234–239.

[5] B. Li *et al.*, "A new cost function for spatial image steganography," in *Proc. 2014 IEEE International Conf. on Image Processing (ICIP)*, 2014, pp. 4206–4210.

[6] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, Jan. 2014.

[7] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2 pp. 221–234, 2015.

[8] T. S. Reinel, R. P. Raul, and I. Gustavo, "Deep learning applied to steganalysis of digital images: A systematic review," *IEEE Access*, vol. 7, pp. 68970–68990, 2019.

[9] R. Tabares-Soto *et al.*, "Digital media steganalysis," in *Digital Media Steganography*, New York: Academic Press, 2020, ch. 1, pp. 259–293.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[11] J. Deng *et al.*, "ImageNet: A large-scale hierarchical image database," in *Proc. 2009 IEEE Conf. on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
[12] P. Bedi, R. Bansal, and P. Sehgal, "Using PSO in a spatial domain based image hiding scheme with distortion tolerance," *Computers & Electrical Engineering*, vol. 39, no. 2 pp. 640–654, 2013.
[13] A. Ahmad, B. M. Ahmad, and S. Y. Lee, "Fast and robust object detection framework in compressed domain," in *Proc. IEEE Sixth International Symposium on Multimedia Software Engineering*, 2004, pp. 210–217.
[14] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—A survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, July 1999.
[15] P. Sallee, "Model-based methods for steganography and steganalysis," *International Journal of Image and Graphics*, vol. 5, no. 01, pp. 167–189, 2005.
[16] S. Dhawan and R. Gupta, "Analysis of various data security techniques of steganography: A survey," *Information Security Journal: A Global Perspective*, vol. 30, no. 9, pp. 1–25.
[17] M. Alanzy *et al.*, "Image steganography using LSB and hybrid encryption algorithms," *Applied Sciences*, vol. 13, no. 21, 11771, 2023.
[18] L. Pibre *et al.*, "Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover source-mismatch," arXiv preprint, arXiv:1511.04855, 2015.
[19] G. Xu, H. Z. Wu, and Y. Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5 pp. 708–712, 2016.
[20] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11 pp. 2545–2557, 2017
[21] M. Yedroudj, F. Comby, and M. Chaumont, "Yedroudj-net: An efficient CNN for spatial steganalysis," in *Proc. 2018 IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2092–2096.
[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint, arXiv:1409.1556, 2014.
[23] B. Patrick, T. Filler, and T. Pevny´, " 'Break our steganographic system': The ins and outs of organizing BOSS," in *Proc. International Workshop on Information Hiding*, 2011, pp. 59–70.
[24] W. Mazurczyk and S. Wendzel, "Information hiding: Challenges for forensic experts," *Communications of the ACM*, vol. 61, no. 1, pp. 86–94, Dec. 2017.
[25] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *Computer*, vol. 31, no. 2, pp. 26–34, Feb. 1998.
[26] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, June 2012.
[27] Y. Qian *et al.*, "Deep learning for steganalysis via convolutional neural networks," in *Proc. Media Watermarking, Security, and Forensics 2015*, 2015, pp. 171–180.
[28] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," in *Proc. the 11th ACM Workshop on Multimedia and Security*, 2009, pp. 75–84.
[29] Y. Qian *et al.*, "Learning and transferring representations for image steganalysis using convolutional neural network," in *Proc. 2016 IEEE International Conf. on Image Processing (ICIP)*, 2016, pp. 2752–2756.
[30] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2018.
[31] R. Zhang *et al.*, "Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1138–1150, 2019.
[32] T. S. Reinel *et al.*, "GBRAS-Net: A convolutional neural network architecture for spatial image steganalysis," *IEEE Access*, vol. 9, pp. 14340–14350, 2021.
[33] J. D. L. C. Ntivuguruzwa and T. Ahmad, "A convolutional neural network to detect possible hidden data in spatial domain images," *Cybersecurity*, vol. 6, 23, Sep. 2023.
[34] A. Martin *et al.*, "Evolving generative adversarial networks to improve image steganography," *Expert Systems with Applications*, vol. 222, 119841, 2023.
[35] A. Darwish and D. Ezzat, "An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis," *Swarm and Evolutionary Computation*, vol. 52, 100616, 2019.
[36] O. C. Do *et al.*, "An efficient approach to medical image fusion based on optimization and transfer learning with VGG19," *Biomedical Signal Processing and Control*, vol. 87, 105370, 2023.
[37] A. A. Mhmood, Ö. Ergül, and J. Rahebi, "Detection of cyber-attacks on smart grids using improved VGG19 deep neural network architecture and Aquila optimizer algorithm," *Signal, Image and Video Processing*, vol. 18, pp. 1477–1491, Nov. 2023.
[38] M. M. Hossain *et al.*, "Particle swarm optimized fuzzy CNN with quantitative feature fusion for ultrasound image quality identification," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 10, pp. 1–12, 2022.
[39] A. M. A. Ahmad and S. Y. Lee, "Fast and robust object-extraction framework for object-based streaming system," *International Journal of Virtual Technology and Multimedia*, vol. 1, no. 1, pp. 39–60, Feb. 2008.
[40] I. A. Albadarneh and A. Ahmad, "Machine learning based oil painting authentication and features extraction," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 17, no. 1, pp. 8–17, 2017.