International Journal of Safety and Security Engineering
Vol. 14, No. 6, December, 2024, pp. 1707-1718
Journal homepage: http://iieta.org/journals/ijsse

Convolutional Neural Network to Detect the Secret Data in the Spatial Domain Images

Bayu Aditya Triwibowo[1], Erick Delenia[1], Yoggy Harisusilo Putra[1], Ntivuguruzwa Jean De La Croix[1,2], Tohari Ahmad[1*]

[1] Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia
[2] College of Science and Technology, University of Rwanda, Kigali 3900, Rwanda

Corresponding Author Email: tohari@its.ac.id

https://doi.org/10.18280/ijsse.140606

Received: 16 December 2023
Revised: 5 July 2024
Accepted: 1 August 2024
Available online: 31 December 2024

Keywords:
CNN, deep learning, information security, national security, network infrastructure, steganalysis

ABSTRACT

Advancements in Deep Learning (DL) have led to innovative approaches to address complex issues, notably steganalysis concerning spatial domain images. Steganalysis is a counter art of steganography that aims to detect the presence of possible hidden data in the pixels of an image. Based on the DL logic, Convolutional Neural Networks (CNNs) have been instrumental in this domain. Over the past years, several CNN architectures have emerged, elevating the accuracy in detecting the images hosting the steganographically hidden data in images. However, existing CNN models face problems associated with limitations in the perceptibility of low payload capacities and less-than-optimal processes for feature learning. This study introduces a novel CNN architecture to enhance the steganalysis process and improve the accuracy of secret data detection for spatial domain images. In the proposed method, the key contributions to CNN development include the utilization of mixed pooling, which combines different pool sizes to enhance the network's ability to capture deeper and multiple shades, thereby providing flexibility in feature extraction. Additionally, depth steganalysis and separable convolution address kernel neglect in channel and residual spatial correlation. Integrating LeakyReLu is proposed to mitigate weak slopes and enhance network convergence. Experimental results demonstrate that employing the proposed CNN architecture improves steganalysis outcomes. It is important to highlight that the findings reveal an accurate improvement of up to 10.2% over the recently considered schemes.

1. INTRODUCTION

The emerging evolution of technology in data transmission has fostered global interactions, with individuals increasingly relying on these interactions to shape their social lives through the public network. People seamlessly transmit and receive diverse data daily through the Internet, particularly in digital media such as audio, video, and images. The possible existence of hidden data in digital media, specifically in images ubiquitous on the public network, poses a risk due to its rapid, widespread, and free distribution, making it susceptible to spreading harmful data [1, 2]. Steganography, the art of data hiding, compounds this issue in digital data distribution, as it can be utilized to transgress upon the privacy of others [3]. While steganography can be embedded in various digital media [4-6], images are the most conducive to this purpose due to their pervasiveness. Employing techniques such as the Least Significant Bit (LSB), Frequency Domain, and CNN-GAN, steganography necessitates a proactive solution, known as steganalysis, to mitigate its occurrence because this approach is mostly used in the transmission of the secret data, which may be harmful to the community [7].

Steganalysis, positioned as the counter-art to steganography, is tasked with detecting possible secret data within digital media, which gets more challenging due to the emergence of steganographic algorithms based on the adaptive logic of the cover image's features [8]. Steganography and steganalysis concepts are inherently intertwined, as steganography's evolution inevitably influences steganalysis's development. Steganalysis employs diverse techniques to identify statistical variances between the original cover image and the steganographic image, known as the stego image, conditional on the specific context in which it operates [9]. Initially, fundamental steganalysis techniques manually extracted statistical characteristics based on human expertise, transitioning to feature extraction methods rooted in pixel correlations. However, the separation between feature extraction and classification processes in these methods hindered their simultaneous optimization, emphasizing the need for iterative enhancement [10]. The persistent evolution of steganography introduces ongoing challenges with increasingly sophisticated designs and algorithms.

Based on the limitations of traditional steganalysis methods, researchers are increasingly turning to Deep Learning (DL) techniques to formulate more effective approaches [11-15]. Successful implementations of Convolutional Neural Networks (CNNs) in various domains underscore their potential in steganalysis, leveraging CNNs' robust capabilities in computer vision [13].

Although DL methods exhibit strong learning capabilities, their training demands extensive data and time. It often yields poor accuracy and results in general, thus prompting continued reliance on manual steganalysis. Using DL models in steganalysis can increase feature dimensions, resulting in significant computational and storage overhead [16].

Existing solutions addressing accuracy issues in steganalysis applications have significantly contributed to achieving high accuracy levels. However, these solutions have several drawbacks, including high computing capacity requirements. Furthermore, the accuracy of these solutions still needs improvement to optimize correctness, especially for sensitive fields like military, medical, and forensic applications.

In response to the identified challenges in existing solutions, this paper introduces an innovative CNN architecture inspired by previous research [12]. The primary goal is overcoming the accuracy challenges of earlier steganalysis methods. Our approach emphasizes memory efficiency and architectural simplicity, introducing a streamlined design that minimizes memory consumption and constrains the overall architecture. The model integrates a pre-processing layer that efficiently applies filters to analyze the input image's pixels. Feature extraction stages combine traditional and depth-wise convolutions with average pooling. Global average pooling is used during the classification phase, followed by a SoftMax function.

Despite the significant advancements in steganalysis using deep learning techniques, existing methods face several critical limitations that reduce their effectiveness and efficiency. Traditional steganalysis techniques often struggle with high computational costs and complex feature extraction processes that are not optimized simultaneously, leading to suboptimal performance. Additionally, deep learning models, while powerful, require extensive datasets and prolonged training times, resulting in increased computational and storage overhead. Furthermore, although achieving high accuracy levels, current solutions demand significant computing capacity and iterative refinement, which are impractical for real-time applications.

The novelty of this paper is highlighted through the following contributions:

(1) Introduction of a design that reduces computational complexity while enhancing accuracy by employing depth-wise separable convolutions and minimizing the number of convolutional layers.

(2) Integration of a pre-processing layer for efficient pixel analysis, which combines traditional and depth-wise convolutions with average pooling to optimize feature extraction and classification processes.

(3) Provides a robust solution for accurate and efficient steganalysis in spatial domain images, addressing the limitations of high computational costs, extensive datasets, and impracticality for real-time applications.

The subsequent sections of this paper are organized as follows: Section 2, "Related Works," provides an overview of the image steganalysis framework and related research. Section 3 outlines the "Proposed Method," presenting our innovative approach. In Section 4, the "Results" section, we detail the experimental setup and obtain and discuss results. Finally, Section 5, the "Conclusion," summarizes our findings and concludes the article.

2. RELATED WORKS

In this section, the literature exploration focuses on steganalysis applied to digital images using DL methods that augment traditional Machine Learning (ML) approaches. The CNN architecture is modified across pre-processing, feature extraction, and classification models to address the cost issue of CNN training. These alterations are informed by several preceding studies that delve into the domain of steganalysis. Over time, the CNN architecture for steganalysis has evolved, with some studies showcasing significant advancements over their predecessors. The development of steganalysis CNN architecture is an ongoing process.

Kang et al. [17] introduced an approach to detecting hidden data in color images by leveraging the correlation between gradient amplitudes of different color channels. By extracting co-occurrence matrix features from these gradient amplitude residuals, the method effectively identifies the weakened correlations that result from color image steganography, demonstrating robust detection capabilities. However, the approach is limited to color images, may face computational challenges due to the intensive feature extraction process, and relies heavily on the quality of training data. Building upon existing research in gradient-based features and co-occurrence matrices, this method enhances the robustness of steganalysis techniques. It provides a comprehensive analysis that could inform future research in the field.

Liu et al. [18] also introduced a traditional approach to steganalysis by applying the Bat Algorithm (BA) for feature selection. The Bat Algorithm, inspired by the echolocation behavior of bats, effectively selects the most relevant features from a large set of candidate features, reducing the risk of overfitting and improving the generalization of steganalysis models. This method is particularly robust and scalable, making it suitable for real-world applications. However, it requires careful parameter tuning and can be computationally intensive. The approach builds upon previous research in feature selection by introducing a more robust and efficient method, enhancing the robustness of steganalysis models against various steganography techniques.

The Xu-Net architecture, initially introduced in previous research [15], integrates conventional Convolutional Neural Network (CNN) components such as batch normalization, global average pooling, and convolutional layers. Its distinctive feature is High-Pass Filtering (HPF) filter banks and Absolute Value (ABS) activation functions during pre-processing. This approach has demonstrated effectiveness in outperforming earlier systems in steganalysis tasks. However, while Xu-Net sets a solid foundation, its reliance on traditional CNN components may limit its adaptability to more complex data distributions in steganography detection.

Previous research [16] presents the Ye-Net architecture following Xu-Net. This architecture enhances feature extraction in stego images by employing channel selection techniques derived from Xu-Net's HPF filters. These filters are transformed into Spatial Rich Models (SRM) filter banks, providing a robust pre-processing step. Introducing the Truncation Linear Unit (TLU) as an activation function marks a significant improvement, resulting in better detection performance. Ye-Net may face scalability and computational efficiency challenges despite these advancements, mainly when processing large datasets.

Yedroudj-Net [19] builds upon the foundational principles of both Xu-Net and Ye-Net, integrating SRM, TLU activation, and batch normalization to enhance its design. Additionally, it introduces data augmentation and adaptive filter banks, which significantly enhance steganography detection capabilities. While Yedroudj-Net shows promising improvements, its complexity may lead to overfitting, especially in scenarios with limited training data. This necessitates careful tuning of hyperparameters to maintain generalization.

Recently, the steganalysis tasks to detect possible hidden data showed the development of many other approaches, such as the ones in the study [20, 21]. As referenced in recent works, Zhu-Net employs two separable convolutional layers, a method inspired by earlier architectures [20]. It aims to improve feature extraction by adding Spatial Pyramid Pooling (SPP) and SRM filter banks. Although Zhu-Net achieves promising results, its classification accuracy still requires enhancement. This limitation highlights the ongoing challenge of balancing model complexity with performance in steganalysis tasks. However, the recent one in the previous research improved the Zhu-Net previously proposed by adding the Spatial Pyramid Pooling (SPP) for feature extraction and SRM filter banks. Promising results were achieved, but classification accuracy required further improvement [20, 21].

Later, in 2021, GBRAS-Net [22] emerged as a significant advancement, drawing inspiration from Zhu-Net. It utilizes SRM filter bank pre-processing under non-trainable conditions and incorporates "skip connections" with depth-wise separable convolutional layers. This architecture has demonstrated superior performance on benchmark datasets such as BOSSBase 1.01 and BOWS, indicating a successful refinement of previous methods. However, the reliance on non-trainable conditions may restrict adaptability in dynamic environments, where model retraining could enhance performance further.

Most recently, the work [12] presented a CNN architecture using depth-wise separable convolution with a skip layer, seven 2D convolution layers, batch normalization, and average pooling layers. This architecture introduced LeakyReLu activation in steganalysis and employed multi-scale average pooling as a classification layer. This study introduced a novel approach to enhance the accuracy of confidential data discovery and stability during the training process for images in the spatial domain. Their method contributed significantly to advancing the field of hidden data detection in spatial domain images, outperforming previous architectures such as GBRAS-Net and Zhu-Net, achieving satisfactory steganography detection. Though the obtained results showed an outperformance in accuracy, a remarkable weakness related to resource consumption due to the heavy network is identified.

In the realm of steganalysis, the development of CNN architectures has significantly contributed to advancing steganography. While adopting commonly used architectural elements, the influence of hyperparameter settings on feature extraction from stego images remains a critical consideration. This article highlights the existing works discussed to propose a new CNN to enhance existing steganalysis models, focusing on convolution operations, encompassing the type and number of layers, and improving computational efficiency during training. The classification phase emphasizes feature modeling and compilation efficiency, ultimately enhancing performance by detecting concealed messages within spatial domain image.

3. PROPOSED METHOD

This research introduces an innovative CNN approach to enhance the detection of concealed information within spatial domain images, thereby advancing steganalysis methodologies. The proposed model addresses the challenges inherent in steganalysis by incorporating insights gleaned from recent breakthroughs in CNN architecture. The method is anticipated to refine the process of image detail extraction through strengthened convolution operations, optimized layer architecture, and improved computational efficiency. Emphasis has also been placed on minimizing the computational effort required during the training phase. These modifications collectively enhance the efficacy of our method in identifying hidden messages within images, thereby contributing to the overall capability of detecting concealed information in spatial domain images. This section provides a detailed exposition of our proposed method's intricacies, elucidates the rationale behind the chosen architecture, and underscores the distinctive features that set our approach apart in the complex realm of steganalysis.

3.1 Architecture

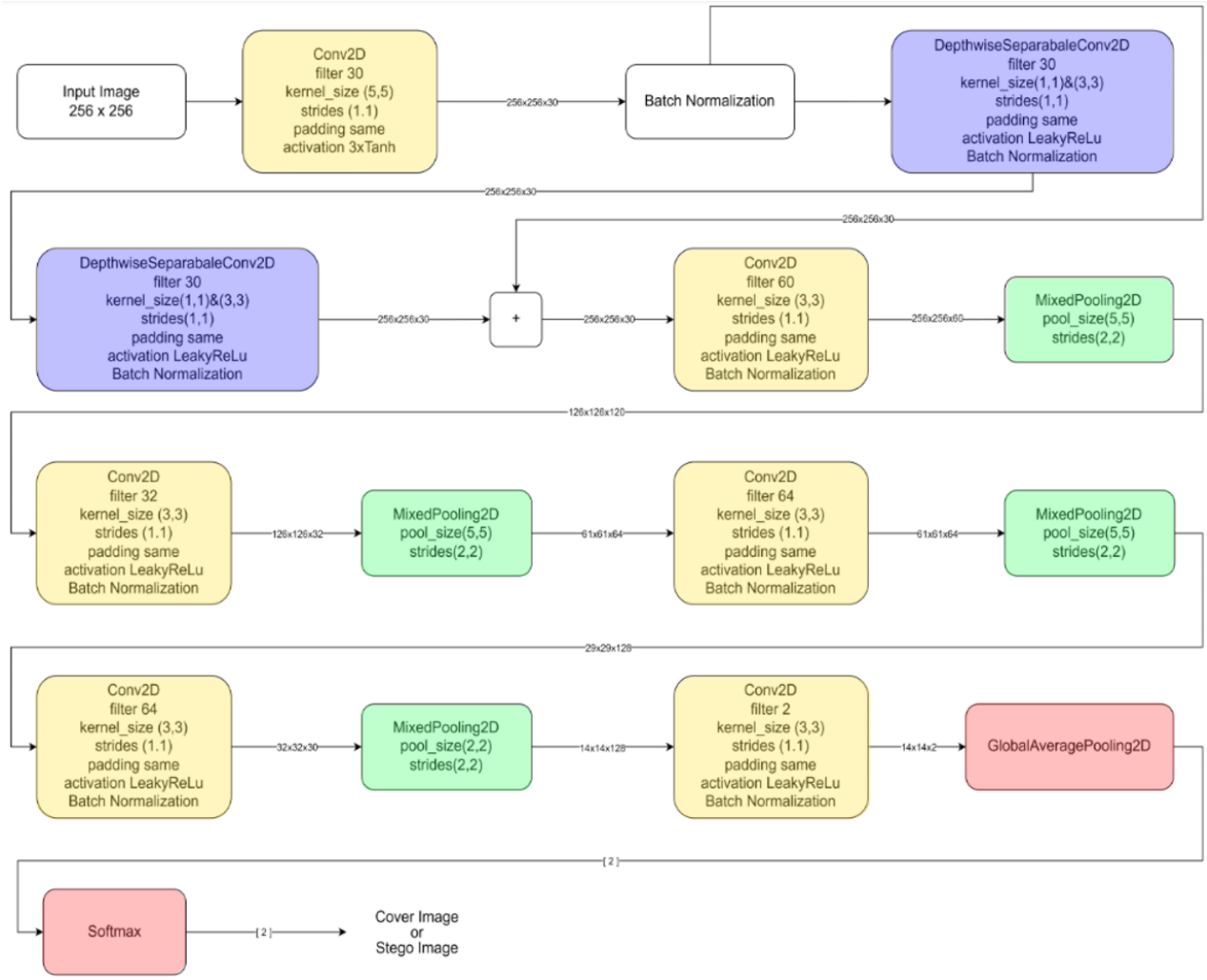
Figure 1 illustrates an overview of the CNN-based steganalysis architecture proposed in this study. CNN is designed to generate two class labels, namely "stego" and "cover," based on an input image with dimensions of 256×256. The architecture comprises various layers, including an image pre-processing layer, two distinct convolution blocks (separable convolution (SepConv)), four basic blocks for feature extraction, mixed pooling, global average pooling, and a final layer with SoftMax activation. The convolution layer extracts spatial correlations between feature maps, which are then transmitted to the classification layer to determine probabilities. Each block within the architecture is defined by a specific set of steps, collectively enabling a comprehensive analysis of the input image. This structured design facilitates extracting features and their subsequent classification by the CNN, ultimately determining whether the image contains stego or cover content.

3.1.1 Convolution layer

This layer conducts a convolution operation on the input, forwarding the outcome to the subsequent layer. The operation entails multiplying weights with the input, resulting in activation. For this investigation, a 3×3 kernel size was chosen, signifying that the filter scans 3×3 blocks of pixels in the image during each convolution step. A relatively modest kernel size, such as 3×3, typically captures local features within the image. The strides (1,1) signify that the kernel moves one pixel to the right and one pixel down at each step. While smaller strides offer a more detailed representation of the image, they may necessitate additional computation time. The filter layers encompass 30, 32, 60, and 64. This assortment facilitates the extraction of multi-level and intricate features corresponding to the hierarchical information levels in the image. Padding, crucial for image edge processing, adopts the 'same' value, signifying the addition of zero pixels around the image to maintain constant dimensions post-convolution. This preserves edge information and mitigates the risk of losing detail in the convolution process. The activation function chosen is the leaky rectified linear unit, introducing a nonlinear function into the model. LeakyReLu permits

gradients on inactive units, addressing the "dying ReLU" issue during training by assigning a small positive value (0.01) to the gradient when the input is less than zero. This enables the model to still learn from less significant information, which is crucial in handling steganographic data with complex variations. A comprehensive consideration of computational complexity and network performance guides determining the number of channels in each layer.

Batch Normalization (BN) is applied following the convolutional operations to enhance the network's performance and stability further. BN normalizes the hidden layer output for each batch of data entering the network, overcoming training instability, accelerating convergence, and reducing sensitivity to weight initialization. By providing stable mean values and standard deviations, BN reduces the effects of internal covariate shifts, i.e., frequent changes in input distribution during training. This stabilization speeds up learning and helps improve the model's generalization ability, reducing overfitting and enhancing the stability of the learning process. This is particularly beneficial for steganalysis, where data can exhibit complex variations. Experimental results indicate that networks without BN are very sensitive to parameter initialization and may fail to converge if initialized improperly. By maintaining the stability and consistency of input values, BN plays a critical role in ensuring the reliability and performance of steganography detection models, balancing the model's performance and stability.



Note: Legend: (1) Orange-colored components represent the used 2D-convolutional layers, which entails multiplying weights with the input, resulting in activation; (2) Blue-colored components represent the used 2D-DepthwiseSeparable convolutions, involving two stages: depth-wise convolution and pointwise convolution; (3) Green-colored components represent the used mixed pooling layers, and the (4) Red-colored components are the ones used in the classification layer.

Figure 1. Proposed CNN architecture

3.1.2 Mixed pooling layer

The Mixed Pooling Layer introduces an innovative approach by integrating different pooling types within a single structure, enabling neural networks to extract feature information more comprehensively. This layer incorporates two distinct types of pooling: Max Pooling and Average Pooling. Max Pooling selects the maximum value from each block or window in the input image during each pooling step, effectively representing the most significant feature within the corresponding pixel block. Conversely, Average Pooling calculates the average value of each block or window in the input image during each pooling step. The advantage of the mixed pooling layer lies in its capacity to capture both prominent and general feature information in the image. The mixing proportion determines the relative contribution of each pooling method to the final output. By integrating both types of pooling, this layer provides a more detailed and contextual representation, enabling the network to learn more effectively from various aspects of steganographic data.

The study implements two distinct pool sizes, (5×5) and (2×2), which offer advantages in the feature extraction hierarchy. The larger pool size (5×5) is adept at capturing more extensive and complex features, while the smaller pool size (2×2) excels at extracting subtle and local features. Combining these pool sizes allows the model to obtain a richer and more diverse image representation, contributing to enhanced feature extraction capabilities. This dual approach enables the model to handle the diverse and intricate characteristics of steganographic data, improving its proficiency in capturing hidden patterns at different levels of detail. The Mixed Pooling Layer, therefore, enhances the model's ability to extract comprehensive and nuanced features, making it a critical component in the steganalysis process.

3.1.3 Depth-wise separable convolution

Depth-wise separable convolution is a pivotal technique in modern CNN architectures [23], involving two stages: depth-wise convolution and pointwise convolution. Depth-wise convolution performs convolution operations on each input channel independently, allowing the model to explore correlations within each channel in isolation. This technique is followed by pointwise convolution (1×1), which combines and enhances the information extracted from the depth-wise convolution stage. The proposed CNN architecture utilizes 1×1 pointwise convolution and 3×3 depth-wise convolution. During depth-wise convolution, each input channel undergoes individual processing, facilitating the extraction of spatial correlations within each channel. This is achieved by using 30 groups in the depth-wise convolution stage. Subsequently, pointwise convolution integrates this information, enhancing feature representation richness and expressiveness.

The combination of these techniques, known as depth-wise separable convolution, offers notable advantages, including a substantial parameter reduction and enhanced computational efficiency. This reduction in computational load results in a lighter network without compromising its representational capacity. Additionally, incorporating skip layers or residual connections in depth-wise separable convolution addresses challenges related to vanishing or exploding gradients during training, enhancing network stability and its ability to learn complex features. Skip layers, commonly applied as shortcut connections, merge information from the previous layer to the next, promoting more effective information flow within the network. By combining depth-wise and pointwise convolutions and integrating skip layers, the model balances computational efficiency and high representational capacity, making it well-suited for tasks like image processing and steganalysis, where detailed feature extraction and efficiency are paramount.

3.1.4 Classification layer

Global Average Pooling (GAP) is a pivotal technique that streamlines the spatial representation of the entire feature map within a CNN. Unlike conventional pooling methods such as Max Pooling or Average Pooling, GAP does not use a dedicated pooling window or kernel. Instead, it globally averages the pixel values in each channel, deriving the average value of each channel over the entire feature map. This global averaging aspect gives GAP its name and underscores its primary advantage: the ability to significantly reduce data dimensionality without sacrificing vital information. By averaging values, GAP fosters a more comprehensive representation of the image or feature from the preceding layer, prioritizing core and essential information. In the research work, the GAP helps focus the Proposed CNN on crucial details while mitigating the risk of overfitting, yielding a representation more invariant to translations and slight variations in object position. This process involves exponential normalization of input values, ensuring the sum of probabilities for all classes equals one. Using GAP and SoftMax enhances computational efficiency and robustness to changes in image size, simplifying feature representation to a single value per channel.

3.2 Hyperparameter selection

The hyperparameters of our CNN architecture, including kernel sizes, the number of filters, and pooling strategies, are selected based on theoretical principles and empirical evidence. We employ a combination of small (3×3) and medium (5×5) kernel sizes. The 3×3 kernel is widely used in image recognition tasks due to its efficiency in capturing fine details and lower computational cost. In contrast, the 5×5 kernel in the pre-processing layer captures broader contextual information, enhancing the detection of steganographic artefacts that may span larger regions. The number of filters, set at 32, 64, and 128 in successive layers, balances the need for capturing sufficient feature representations while maintaining computational efficiency. This gradual filter increase aligns with the hierarchical nature of feature learning in CNNs, where initial layers capture basic features and deeper layers learn more complex patterns.

Pooling strategies are crucial for reducing spatial dimensions and mitigating overfitting. We use average pooling instead of max pooling because it retains more spatial information, which is essential for distinguishing between cover and stego images. Additionally, global average pooling is employed before the final classification layer to aggregate feature maps robustly for the SoftMax classifier. These choices are guided by empirical studies showing that average pooling enhances performance in tasks requiring detailed spatial analysis [24]. Combining depth-wise separable convolutions and minimizing the convolutional layers further reduces computational complexity and mitigates overfitting, providing a practical and effective solution for steganalysis. Our model balances accuracy, efficiency, and computational feasibility by integrating these well-justified hyperparameters.

3.3 Benchmarking the proposed method

The proposed architecture undergoes a comparative analysis with cutting-edge steganalysis architectures highlighted in the preceding section. Specifically, this method is juxtaposed with the feature network structures of GBRAS-Net and Zhu-Net. This selection is motivated by their relevance and prominence in CNN-based steganalysis development, showcasing performance on par with their predecessors in the literature. The input image dimensions for both GBRAS-Net and Zhu-Net are standardized at 256×256, which we also adopted in this research. In the pre-processing stage, Zhu-Net utilizes a filter bank comprising 30 SRM. At the same time, GBRAS-Net employs the same filter bank as a non-trainable filter. Our approach aligns with GBRAS-Net, utilizing 30 SRM as a non-trainable filter with a kernel size of 5×5.

Divergences arise in convolutional layers: Zhu-Net employs 5, GBRAS-Net employs 9, and our research incorporates six convolutional layers. Additionally, variations exist in the utilization of depth-wise separable convolutional layers. Zhu-Net employs two separable conv layers, GBRAS-Net uses four separable and four depth layers, and our research adopts two separable and two depth layers, halving the GBRAS-Net setup for computational efficiency. We introduce a skip layer to mitigate kernel issues and enhance the training process, particularly concerning SNR.

All convolutional layers in the proposed architecture employ LeakyReLu as the nonlinear activation function, chosen for its capacity to improve feature extraction. In contrast, Zhu-Net uses ReLU, and GBRAS-Net employs ELU. In the final layer for the classification process, Zhu-Net incorporates multi-scale mean pooling and connects it to two dense layers with SoftMax for probability determination. GBRAS-Net opts for global average pooling and SoftMax as layers for the classification phase. Our approach aligns with GBRAS-Net, with the distinction that our process integrates mixed pooling extensively in the feature extraction phase, subsequently fed into global average pooling as a determinant of probability results.

4. RESULTS

This section unfolds the outcomes of this research, with a primary focus on the experimental setup, results, and discussion, accompanied by a comparative analysis of the proposed model results against prior research. The experimental setup constitutes the foundation of the investigation, delineating the methodology employed for data collection in intricate detail.

4.1 Experimental setup

4.1.1 Dataset

This section details the dataset utilized in our study, sourced from Break Our Steganographic System Base 1.01 (BOSSBase 1.01) [25]. The dataset comprises 10,000 spatial domain or grayscale images, each with a dimension of 512×512 pixels, originating from seven distinct cameras. BOSSBase 1.01's widespread use in steganalysis research facilitates meaningful comparisons with previous studies. Two steganography algorithms, S-UNIWARD and WOW, configured with a payload of 0.4bpp, were tested on this dataset.

Due to computational considerations and in alignment with certain prior research, we opted to resize the images to 256×256 in this study. The dataset processing, including the image compression for secret message embedding, was executed using MATLAB. Our experiment fully employed this dataset as training, validation, and test data. Out of the 10,000 processed images, designated as stego images, a 50:50 division ensures 20,000 image data, evenly split between cover and stego images. Our approach aligns with established precedent for the training, testing, and validation data division. Specifically, 4,000 covers and 4,000 stego images constitute the training data, 1,000 covers and 1,000 stego images from the validation set, and the remaining data is allocated for the test set.

4.1.2 Computational resource

This study was conducted on a restrained scale, with Google Colaboratory as the primary computational resource for executing Convolutional Neural Network (CNN) computations. The CNN model was run on an NVIDIA T4 GPU boasting 15 GB of dedicated GPU RAM. Concurrently, the dataset processing tasks were undertaken using MATLAB on a computer system featuring an AMD Ryzen 7 4800H CPU configuration and 16 GB of RAM. This hybrid setup was deliberately chosen to capitalize on the parallel processing capabilities of the GPU for resource-intensive CNN computations. At the same time, the robust CPU configuration adeptly managed dataset manipulations. The collaborative use of Google Colaboratory and the local computing facility, each equipped with distinct hardware specifications, was pivotal in establishing a harmonious and efficient workflow, ensuring a balanced approach throughout our research endeavors.

4.1.3 Hyperparameter tuning

In the proposed scheme, we have implemented a batch of size 32 during the CNN training with 100 epochs for a given payload. The epsilon and momentum norms are set to 0.001 and 0.4, respectively, while the batch normalization is set to 0.2. The convolutional layers use the Glorot normal initialization with a kernel size of 3×3 and several filters and adopt the LeakyReLu activation. The applied learning rate is 0.001, while the gradient value of LeakyReLu equals 0.1.

4.1.4 Metrics evaluation

This study employs the accuracy (ACC) calculated using (1) as a primary evaluation metric, serving as a foundational benchmark to assess the performance of the proposed approach. Accuracy as the principal metric aims to deliver a comprehensive evaluation of the overall effectiveness of the proposed approach. Moreover, accuracy facilitates direct comparisons with previous research, fostering a unified understanding of the advancements or distinctions in the proposed methodology compared to existing studies. The consistent application accuracy ensures a standardized evaluation framework, enhancing the effectiveness of the analysis in assessing the proposed model and laying the groundwork for subsequent discussions regarding the implications of our findings in a broader scientific context.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (1)$$

Moreover, the proposed method is evaluated using precision (PRC) as Eq. (2), Recall (REC) obtained from Eq. (3), and F1-Score computed using Eq. (4). Precision assesses the proportion of correct positive predictions among all positive predictions, offering insights into the model's ability to avoid false positive predictions. Conversely, Recall evaluates the model's capacity to capture all true positive instances by assessing the ratio of true positives to the sum of true positives and false negatives. The F1-Score combines precision and Recall, providing a metric to measure false positives and negatives, especially in scenarios where a balance between precision and Recall is crucial. Including precision, Recall, and F1 scores offer a more nuanced perspective on the model's performance, enabling the identification of specific strengths and weaknesses in its predictive ability.

$$PRC = \frac{TP}{TP + FP} \times 100\% \quad (2)$$

$$REC = \frac{TP}{TP + FN} \times 100\% \quad (3)$$

$$F1 - Score = 2 \times \frac{PRC \times REC}{PRC + REC} \quad (4)$$

4.2 Results and discussion

This section comprehensively analyzes the achieved conditions to elucidate the system's accuracy in detecting hidden data. Furthermore, comparisons with existing research benchmarks highlight the advancements introduced by our methodology. The subsequent section delves into the results within the broader context of steganalysis, outlining potential avenues for improvement and future research directions. The training and validation results of our CNN architecture are depicted in Figure 2, offering valuable insights into the performance of the S-UNIWARD and WOW steganography algorithms. Notably, the model optimized for the S-UNIWARD algorithm exhibits a high training accuracy of 95.82% but a substantial drop in validation accuracy to 83.77%. This discrepancy indicates overfitting, where the model excels in learning the training data but struggles to generalize patterns to unseen data. The disparity between training loss (0.30) and validation loss (0.40) in the S-UNIWARD model suggests excessive complexity, hindering its adaptability to variations in the validation data.

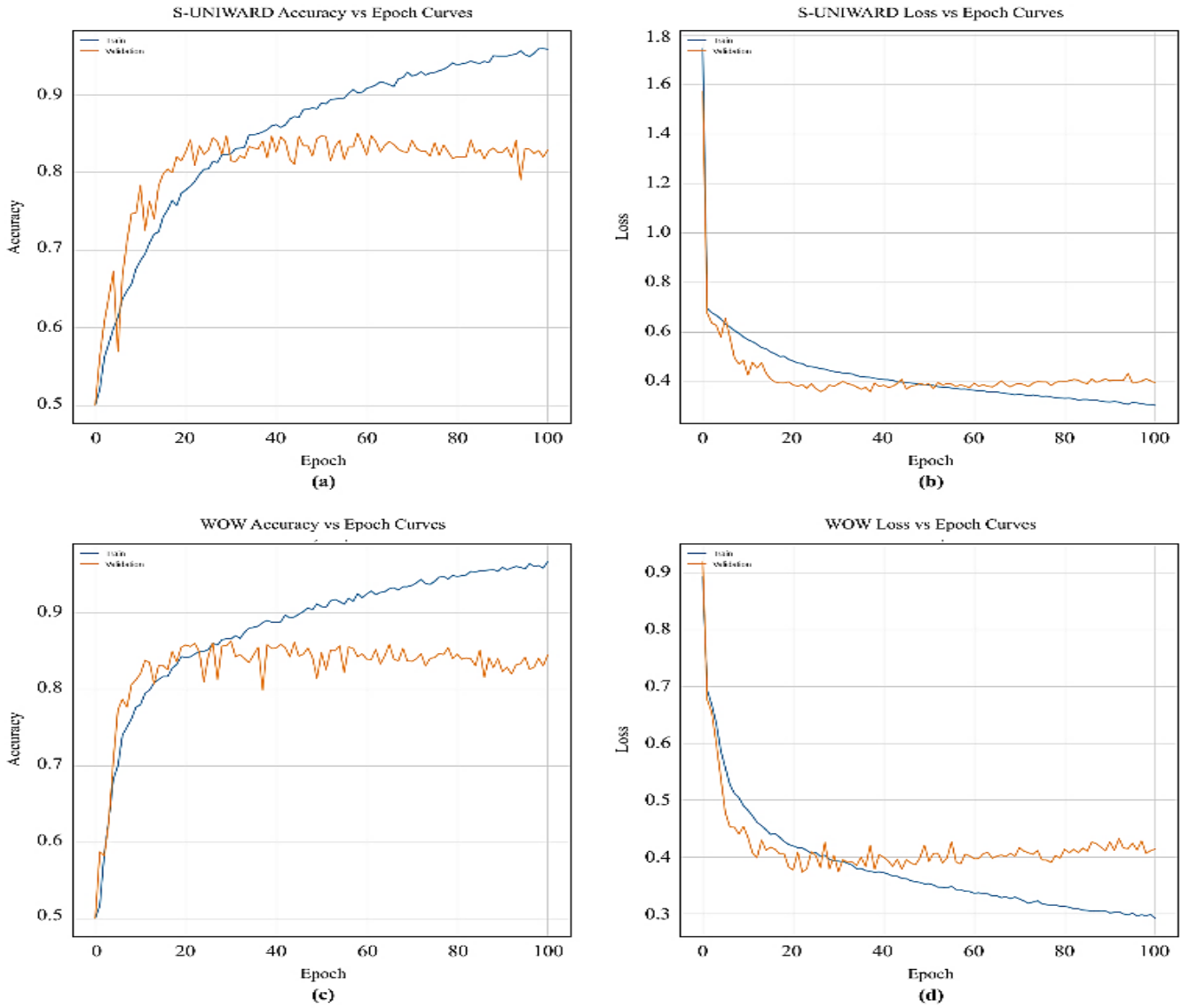


Figure 2. Accuracy and Loss Curves (a) S-UNIWARD Accuracy (b) S-UNIWARD Loss (c) WOW Accuracy (d) WOW Loss

Conversely, the model optimized for the WOW algorithm displays better results, achieving high accuracy rates in both training (96.71%) and validation (86.29%) stages. However, a noticeable difference between training loss (0.30) and validation loss (0.41) hints at potential overfitting, albeit less pronounced than the S-UNIWARD model. While the training results showcase the model's proficiency in understanding patterns within the training data, weaknesses emerge in its ability to generalize to unseen data, particularly in the S-UNIWARD-optimized model. Future research recommendations encompass exploring CNN architecture, parameter tuning, and implementing regularization techniques to mitigate overfitting. Additionally, utilizing larger and more

diverse datasets is advised to enhance the model's adaptability to various steganalysis scenarios.

Further scrutiny of the training and validation data holds promise for additional insights to enhance the model's reliability and generalizability. The strength of the training results is the model's ability to understand and learn the patterns in the training data, as reflected in the high accuracy in the training stage. However, weaknesses arise in the model's ability to generalize to unseen data, especially in the model optimized for the S-UNIWARD algorithm.

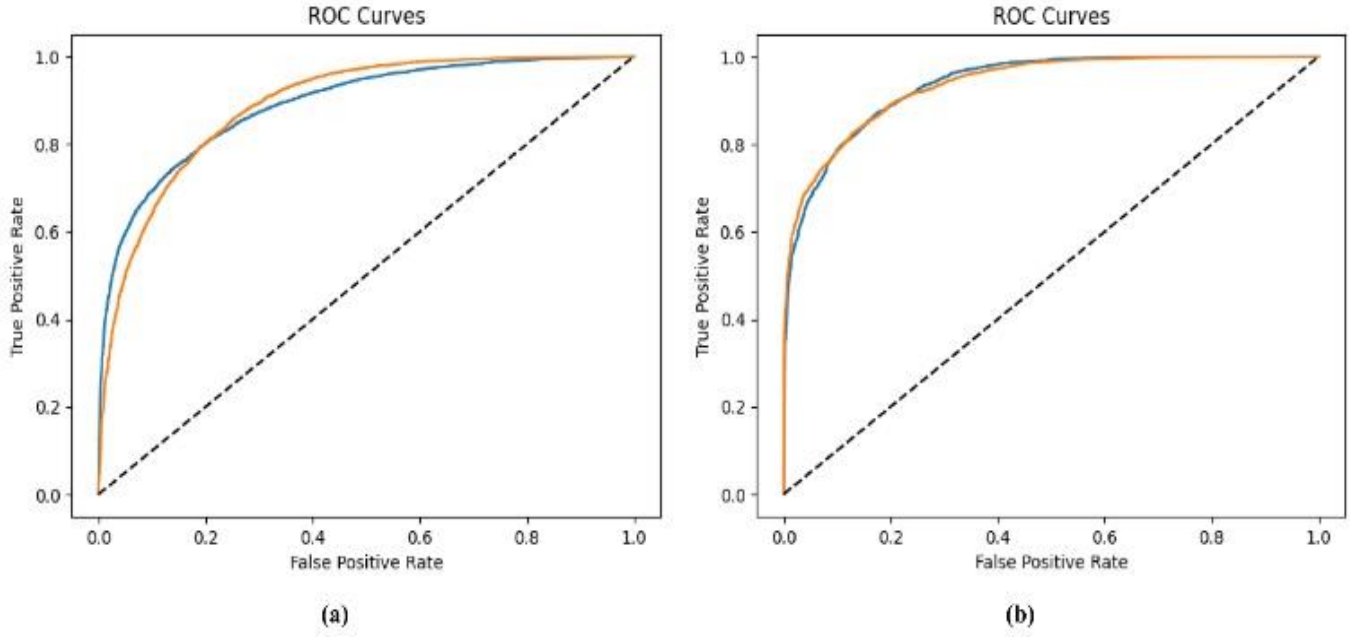


Figure 3. ROC curves (a) S-UNIWARD (b) WOW

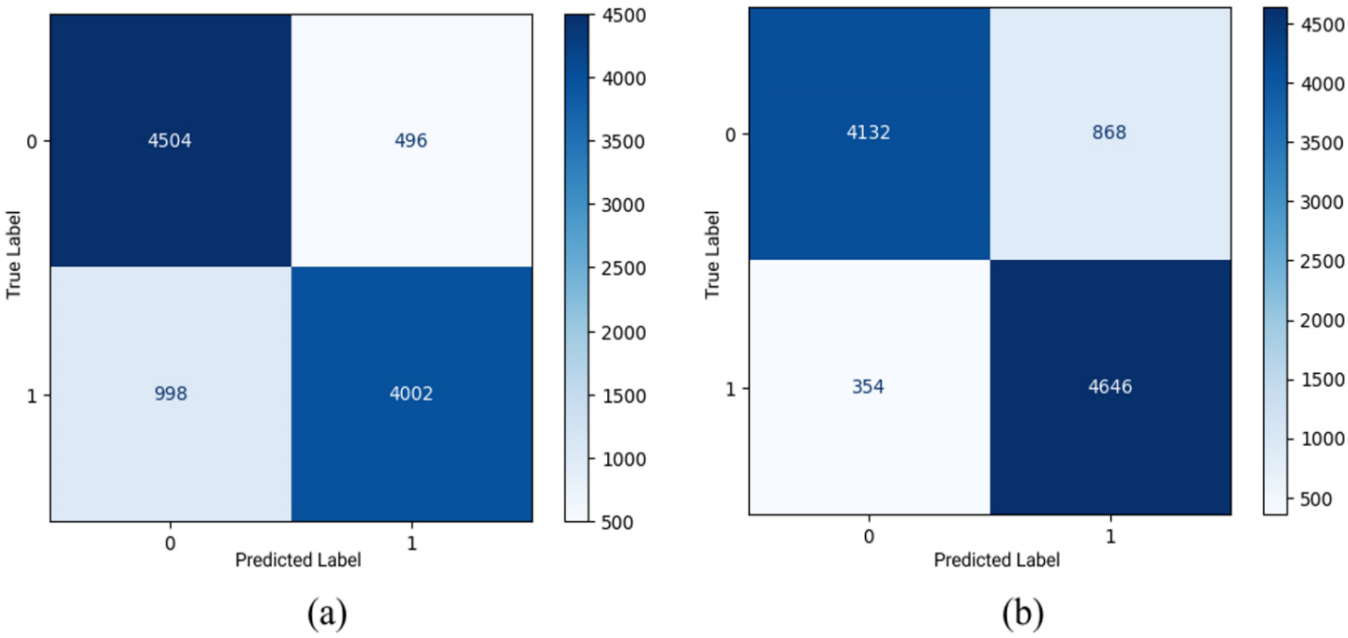


Figure 4. Confusion matrix (a) S-UNIWARD (b) WOW

Table 1. ACC, PRC, REC, and FOS for the proposed CNN

Algorithm	ACC	PRC	REC	FOS
S-UNIWARD	83.7	84.1	83.6	83.8
WOW	86.4	86.4	86.4	86.4

The data in Table 1 include steganalysis classification results obtained by attacking the two commonly known adaptive steganography algorithms, S-UNIWARD and WOW, based on several key evaluation metrics: the ACC, the PRC, the REC, and the F1-Score. Overall, the WOW algorithm achieved an accuracy rate of 86.4%, slightly higher than S-UNIWARD's 83.7%. These two algorithms do tend to have such characteristics. In many other studies, the trend has been the same. However, it should be noted that accuracy alone does not cover all aspects of model performance. In addition, precision and Recall provide a more in-depth picture of the model's ability to identify stego and cover images.

Regarding precision, the S-UNIWARD algorithm scored slightly lower (84.1%) than WOW (86.4%). The high precision indicates that S-UNIWARD is more likely to make

correct positive predictions when the model states that an image is a stego image. On the other hand, despite its high precision, the WOW algorithm also needs to be balanced with other considerations. However, when looking at Recall, the S-UNIWARD algorithm faces a relatively low value (83.6%).

In comparison, the WOW algorithm shows a recall of 86.4%. The low Recall indicates that both algorithms tend to miss many actual stego images and, therefore, may be less sensitive in detecting images that have been modified. The F1-Score, a balanced measure of precision and Recall, provides an overall picture of the performance balance between the two algorithms. Although the WOW algorithm has a slightly lower F1 Score, it may show a slight decrease in the balance between precision and Recall compared to S-UNIWARD.

Table 2. Best epoch based on validation accuracy

S-UNIWARD		WOW	
Val_acc	**Epoch**	**Val_acc**	**Epoch**
84.40	**21**	86.80	**43**
84.30	30	86.80	59
84.05	18	86.15	55
83.95	33	86.00	42
83.89	36	86.00	60

Table 2 identifies epochs that have yielded the best validation accuracy and crucial information in optimizing training duration. Identifying the optimal epochs aids in achieving peak accuracy efficiently, potentially minimizing the need for extensive training. For S-UNIWARD, the optimal epoch falls within the 20-40 range. Similarly, the sweet spot lies between epochs 40-60 for WOW. Researchers can make informed decisions by recognizing these optimal epoch ranges and striking a balance between model performance and computational costs.

The results of the proposed method are also visualized through the ROC curves in Figure 3. The S-UNIWARD model has a slight shift in the Stego and Cover classes. Then, the results of the AUC of the S-UNIWARD model are 0.89, and for the WOW model, they are 0.91. These results can provide insight into the extent to which the model can distinguish between positive and negative classes. Generally, these results illustrate the trade-off between different evaluation metrics, requiring careful consideration when choosing an algorithm based on specific needs and priorities in steganalysis.

In addition to ROC Curves, the significance of the Proposed Method is further highlighted using a confusion matrix that displays the map of model prediction results on Stego and Cover images. Both SUNIWARD and WOW algorithms are mapped into the confusion matrix, as shown in Figure 4. The interesting result of this confusion matrix is that the model tends to predict "label 0" or "cover" in the SUNI algorithm, while it tends to predict "label 1" or "stego" in the WOW algorithm. However, the precision for the SUNI algorithm is 0.82 and 0.85 for label "0" and label "1," respectively, while for the WOW algorithm, it is 0.90 and 0.84 for label "0" and label "1", respectively. The tendency of prediction on one of the labels does not mean high precision either, but thus, the correct or correctly predicted data is also high. This is evidenced by the higher Recall and f1 score values on the labels that tend to be chosen. This confusion matrix can help further explore and understand the CNN model built.

4.3 Results comparison with the existing methods

This subsection compares the accurate results of four significant steganalysis models, Yedroudj-Net, Zhu-Net, GBRAS-Net, and Proposed, each using two steganography algorithms, S-UNIWARD and WOW, as presented in Table 3. This comparative analysis compares these models' performance detecting stego images using different steganography techniques. We compared it with traditional and deep learning methods. We can identify the most effective model and algorithm combinations in steganalysis by exploring the differences in accuracy results. In-depth analysis of the performance comparison between four steganalysis models, namely Yedroudj-Net, Zhu-Net, GBRAS-Net, and our Proposed Method, by applying the S-UNIWARD and WOW algorithms with payload 0.4 bpp, reveals notable improvements in evaluation metrics, particularly accuracy.

Compared to traditional methods such as Liu and Kang's, the results show a significant difference in accuracy, reaching 22.33% on the WOW dataset. Liu's Bat Algorithm reached 64.07% for its accuracy. Of course, this is possible due to advanced deep learning. The difference in ability that occurs causes a high-value gain. The Proposed Model demonstrates superior accuracy improvements.

Table 3. Accuracy comparison between the existing methods and the proposed CNN

CNN	S-UNIWARD	WOW
Yedroudj-Net	77.2	84.1
Zhu-Net	80.1	84.4
GBRAS-Net	81.4	85.9
Proposed	**83.7**	**86.4**

Table 3 shows that the Proposed model attains remarkable accuracy, with the superior values highlighted in bold characters, reaching 83.7% using the S-UNIWARD algorithm and 86.4% with WOW. These exceptional results surpass the performance of other models across both algorithms, including Yedroudj-Net, Zhu-Net, and GBRAS-Net. This underscores the effectiveness of our Proposed Method in steganalysis, marking a significant stride in achieving superior accuracy compared to contemporary models. In model comparisons with Yedroudj-Net, a notable enhancement is observed for the Proposed model. The accuracy of the Proposed model witnessed an improvement of 6.5% with S-UNIWARD and 2.3% with WOW.

Similarly, compared to Zhu-Net, the Proposed model demonstrates an improvement of 3.6% (S-UNIWARD) and 1.5% (WOW). Against GBRAS-Net, a recorded improvement of 2.3% (S-UNIWARD) and 0.5% (WOW) further underscores its superiority. This meticulous analysis highlights the Proposed model's effectiveness in stego image detection and provides profound insights into its enhanced performance with specific algorithm combinations. The Proposed model distinguishes itself by implementing 30 SRM filter banks, LeakyReLu functions, and mixed pooling, showcasing potential feature extraction and steganography detection advantages. However, the model's complexity introduces challenges such as increased computational demands and potential overfitting due to extensive feature extraction.

Additionally, while the Proposed Model excels in the evaluated benchmarks, its generalization to other steganographic domains and scalability in practical applications need further validation.

Overall, the Proposed Model's sophisticated architecture offers substantial improvements in accuracy and versatility but requires careful consideration of its computational and generalization limitations.

This detailed analysis demonstrates that the proposed model offers a more efficient solution for detecting stego images, showing consistent and substantial improvements across various evaluation metrics.

The integrated approach, incorporating nonlinear functions from filter banks, such as TLU and ReLU, along with feature normalization using BN, yields impressive results in steganalysis.

However, the proposed method still requires improvements in the model's architecture to optimize training time, which remains a challenge common to existing works.

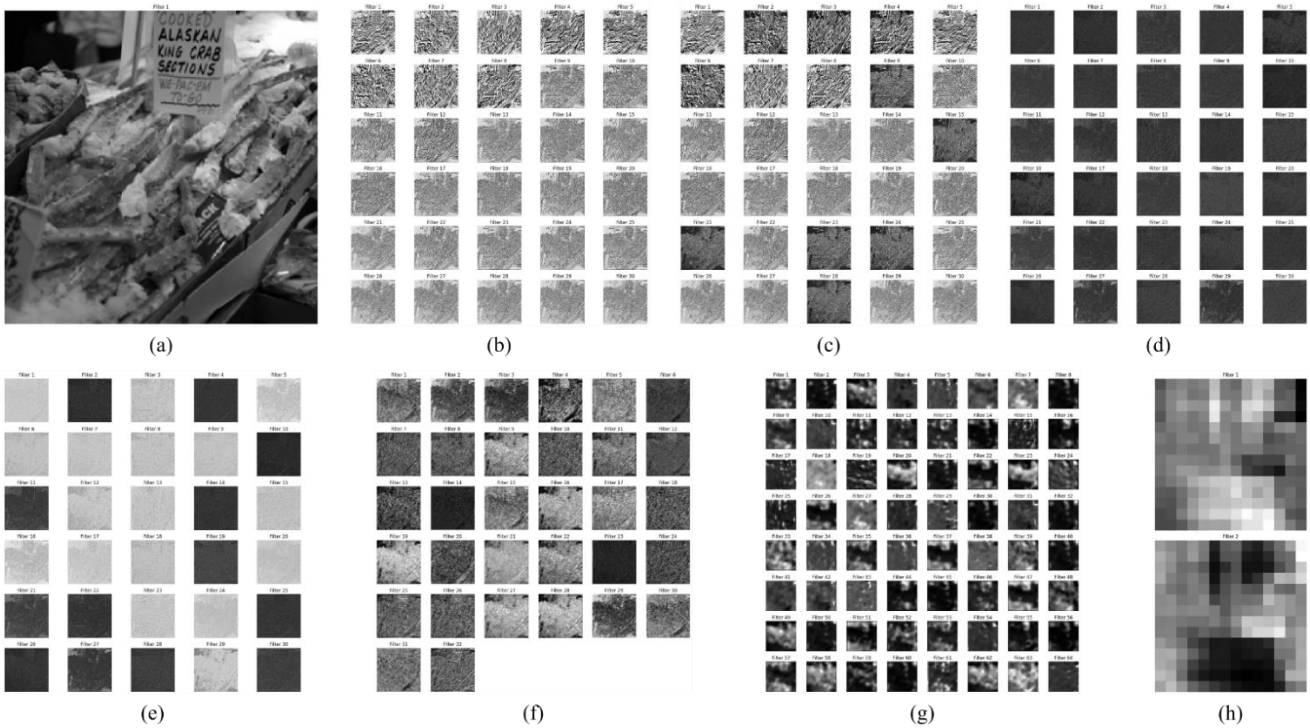


Figure 5. Feature map of (a) Input layer (b) 1st Conv2D layer (c) 1st depthwise layer (d) 1st separable layer (3) 2nd depthwise (f) 3rd Conv2D layer (g) 4th Conv2D layer (h) Last Conv2D layer

4.4 Model's learned features

To understand the model in more detail, we visualize our proposed method's output layer or so-called feature map, which can be seen in Figure 5. In the given image, we observe the progression of feature maps from layer to layer. Initially, after the first 2D convolutional layer, the feature map highlights basic patterns such as edges and lines. These fundamental features serve as building blocks for more intricate representations. Subsequently, the depthwise and separable layers contribute to the abstraction process. Here, features become less recognizable but capture higher-level characteristics, textures, and context. Finally, the last 2D convolutional layer produces highly processed feature maps essential for subsequent classification tasks. These maps encode crucial information about the input image, enabling the network to make informed decisions. This gradual process allows the Proposed model to understand visual information step by step and acquire relevant feature representations

4.5 Potential applications

The proposed CNN-based steganalysis method has significant practical implications and offers various potential applications in digital forensics, cybersecurity, and privacy protection. In digital forensics, this method plays a critical role by uncovering hidden data within images, serving as crucial evidence in criminal investigations, and helping to maintain the chain of custody by verifying image integrity. In cybersecurity, the method is a powerful shield against data leakage by adeptly detecting hidden data transmissions in corporate and network environments, enhancing antivirus solutions, and identifying steganographic malware.

Furthermore, the proposed model supports privacy protection goals by detecting unauthorized surveillance attempts, uncovering embedded spy software within images, and ensuring compliance with data protection laws by preventing unauthorized data transmission. These capabilities make the steganalysis method indispensable for law enforcement agencies, corporate security teams, and individuals seeking to safeguard sensitive information. Notably, its efficiency and effectiveness balance accuracy with computational feasibility, making it well-suited for real-time applications.

Beyond the technological aspects, steganalysis also considers its social impact. Evaluating the societal benefits and challenges associated with steganalysis tools is essential, as these tools address critical security needs. Ultimately, their practical value lies in their application during cybercrime investigations, where they unveil hidden data and enhance forensic analysis, thereby addressing specific cybersecurity threats.

5. CONCLUSION

In the realm of advancing steganalysis methodologies, this research has been dedicated to enhancing performance through the application of a CNN approach. The proposed method not only surpasses existing techniques in detection accuracy but also demonstrates superiority, particularly in comparison with predecessor architectures, notably on the S-UNIWARD and WOW payload 0.4bpp steganography algorithms. The principal contributions to CNN development lie in the innovative utilization of mixed pooling techniques, providing flexibility in feature extraction by combining different pool sizes to capture deeper and more diverse nuances of images. Furthermore, incorporating depth-wise steganalysis and separable convolution prevents kernel neglect in channel correlation, ensuring a robust consideration of spatial correlation. The activation function LeakyReLu is adopted to address the issue of dim gradients, thereby enhancing network convergence. This research affirms the effectiveness of employing CNNs to address the steganalysis problem.

Recommendations for future research include further exploration of CNN architecture, parameter tuning, and the application of regularization techniques to overcome overfitting. In addition, consideration should be given to using larger and more varied datasets so that the model can learn from various situations that may arise in steganalysis. Further analysis of the training and validation data may also provide additional insights to improve the reliability and generalizability of the model.

ACKNOWLEDGMENT

The authors gratefully acknowledge support from the Institut Teknologi Sepuluh Nopember for this work, under project scheme of the Publication Writing and IPR Incentive Program (PPHKI) 2024.

REFERENCES

[1] Rana, K., Singh, G., Goyal, P. (2023). SNRCN2: Steganalysis noise residuals-based CNN for source social network identification of digital images. Pattern Recognition Letters, 171: 124-130. https://doi.org/10.1016/j.patrec.2023.05.019

[2] Yin, Z., She, X., Tang, J., Luo, B. (2021). Reversible data hiding in encrypted images based on pixel prediction and multi-MSB planes rearrangement. Signal Processing, 187: 108146. https://doi.org/10.1016/j.sigpro.2021.108146

[3] Fidler, D.P. (2012). Tinker, Tailor, Soldier, Duqu: Why cyberespionage is more dangerous than you think. International Journal of Critical Infrastructure Protection, 5(1): 28-29. https://doi.org/10.1016/j.ijcip.2011.12.001

[4] Su, W., Ni, J., Hu, X., Huang, F. (2022). Towards improving the security of image steganography via minimizing the spatial embedding impact. Digital Signal Processing, 131: 103758. https://doi.org/10.1016/j.dsp.2022.103758

[5] Mahmoud, M.M., Elshoush, H.T. (2022). Enhancing LSB using binary message size encoding for high capacity, transparent and secure audio steganography–An innovative approach. IEEE Access, 10: 29954-29971. https://doi.org/10.1109/ACCESS.2022.3155146

[6] Martini, M. (2023). A simple relationship between SSIM and PSNR for DCT-based compressed images and video: SSIM as content-aware PSNR. 2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP), pp. 1-5. https://doi.org/10.1109/MMSP59012.2023.10337706

[7] Li, N., Qin, J., Xiang, X., Tan, Y. (2023). Robust coverless video steganography based on inter-frame keypoint matching. Journal of Information Security and Applications, 79: 103653. https://doi.org/10.1016/j.jisa.2023.103653.

[8] Fu, Z., Chai, X., Tang, Z., He, X., Gan, Z., Cao, G. (2024). Adaptive embedding combining LBE and IBBE for high-capacity reversible data hiding in encrypted images. Signal Processing, 216: 109299. https://doi.org/10.1016/j.sigpro.2023.109299

[9] Fu, T., Chen, L., Fu, Z., Yu, K., Wang, Y. (2022). CCNet: CNN model with channel attention and convolutional pooling mechanism for spatial image steganalysis. Journal of Visual Communication and Image Representation, 88: 103633. https://doi.org/10.1016/j.jvcir.2022.103633

[10] Gupta, S., Mohan, N., Kaushal, P. (2022). Passive image forensics using universal techniques: A review. Artificial Intelligence Review, 55(3), 1629-1679. https://doi.org/10.1007/s10462-021-10046-8

[11] Tang, W., Li, B., Tan, S., Barni, M., Huang, J. (2019). CNN-based adversarial embedding for image steganography. IEEE Transactions on Information Forensics and Security, 14(8): 2074-2087. https://doi.org/10.1109/TIFS.2019.2891237

[12] Ntivuguruzwa, J.D.L.C., Ahmad, T. (2023). A convolutional neural network to detect possible hidden data in spatial domain images. Cybersecurity, 6(1): 23. https://doi.org/10.1186/s42400-023-00156-x

[13] De La Croix, N.J., Ahmad, T., Han, F. (2024). Comprehensive survey on image steganalysis using deep learning. Array, 22: 100353. https://doi.org/10.1016/j.array.2024.100353

[14] Alrubaie, H.D., Aljobouri, H.K., Aljobawi, Z.J. (2023). Efficient feature selection using CNN, VGG16 and PCA for breast cancer ultrasound detection. Revue d'Intelligence Artificielle, 37(5): 1255-1261. https://doi.org/10.18280/ria.370518

[15] Xu, G., Wu, H.Z., Shi, Y.Q. (2016). Structural design of convolutional neural networks for steganalysis. IEEE Signal Processing Letters, 23(5): 708-712. https://doi.org/10.1109/LSP.2016.2548421

[16] Ye, J., Ni, J., Yi, Y. (2017). Deep learning hierarchical representations for image steganalysis. IEEE Transactions on Information Forensics and Security, 12(11): 2545-2557. https://doi.org/10.1109/TIFS.2017.2710946

[17] Kang, Y., Liu, F., Yang, C., Xiang, L., Luo, X., Wang, P. (2019). Color image steganalysis based on channel gradient correlation. International Journal of Distributed Sensor Networks, 15(5). https://doi.org/10.1177/1550147719852031

[18] Liu, F., Yan, X., Lu, Y. (2020). Feature selection for image steganalysis using binary bat algorithm. IEEE

Access, 8: 4244-4249. https://doi.org/10.1109/ACCESS.2019.2963084

[19] Yedroudj, M., Comby, F., Chaumont, M. (2018). Yedroudj-net: An efficient CNN for spatial steganalysis. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, pp. 2092-2096. https://doi.org/10.1109/ICASSP.2018.8461438

[20] Zhang, R., Zhu, F., Liu, J., Liu, G. (2019). Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis. IEEE Transactions on Information Forensics and Security, 15: 1138-1150. https://doi.org/10.1109/TIFS.2019.2936913

[21] Kato, H., Osuge, K., Haruta, S., Sasase, I. (2020). A preprocessing by using multiple steganography for intentional image downsampling on CNN-based steganalysis. IEEE Access, 8: 195578-195593. https://doi.org/10.1109/ACCESS.2020.3033814

[22] Reinel, T.S., Brayan, A.A.H., Alejandro, B.O.M., Alejandro, M.R., Daniel, A.G., Alejandro, A.G.J., Buenaventura, B.J.A., Simon, O.A., Gustavo, L., Raul, R.P. (2021). GBRAS-Net: A convolutional neural network architecture for spatial image steganalysis. IEEE Access, 9: 14340-14350. https://doi.org/10.1109/ACCESS.2021.3052494

[23] De La Croix, N.J., Ahmad, T. (2024). A scheme based on convolutional neural network and fuzzy logic to identify the location of possible secret data in a digital image. International Journal on Engineering Applications, 12(1): 1-14. https://doi.org/10.15866/irea.v12i1.23475

[24] Qin, Z., Yu, F., Liu, C., Chen, X. (2018). How convolutional neural networks see the world: A survey of convolutional neural network visualization methods. Mathematical Foundations of Computing, 1(2): 149-180. https://doi.org/10.3934/mfc.2018008

[25] Bas, P., Filler, T., Pevný, T. (2011). Break our steganographic system: The ins and outs of organizing BOSS, Part of the Book Series: Lecture Notes in Computer Science, pp. 59-70. https://doi.org/10.1007/978-3-642-24178-9_5