International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 29, No. 4, August, 2024, pp. 1653-1665
Journal homepage: http://iieta.org/journals/isi

Sensitivity of a Convolutional Neural Network for Different Pooling Layers in Spatial Domain Steganalysis

Yoggy H. Putra[1], Bayu A. Triwibowo[1], Erick Delenia[1], Ntivuguruzwa Jean De La Croix[1,2], Tohari Ahmad[1*]

[1] Department of Informatics, Institut Teknologi Sepuluh Nopember, Kampus ITS, Surabaya 60111, Indonesia
[2] African Centre of Excellence in Internet of Things, University of Rwanda, Kigali 3900, Rwanda

Corresponding Author Email: tohari@its.ac.id

https://doi.org/10.18280/isi.290438

Received: 17 December 2023
Revised: 9 July 2024
Accepted: 29 July 2024
Available online: 21 August 2024

Keywords:
CNN, information security, national security, pooling layer, spatial domain images, steganalysis

ABSTRACT

In the modern era, numerous research studies consistently affirm the superior performance of Convolutional Neural Networks (CNNs) over traditional machine learning methods in steganalysis, a technique used to detect hidden data through steganography. Deep Learning (DL), particularly CNNs, is a powerful tool for steganalysis because it can handle large datasets effectively. Despite CNNs being widely used in various research areas, previous steganalysis studies have primarily focused on improving image classification (cover or stego), often neglecting a thorough exploration of the experimental setup. This research aims to assess the sensitivity of a CNN-based steganalysis model by investigating the impact of different pooling layers on state-of-the-art models. The experiments involve five recently proposed models. Significantly, the choice of pooling layers goes beyond mere classification improvement; it also addresses overfitting. The experimental results reveal significant diversity based on the selected pooling layers, namely the maximum, average, and mixed pooling, emphasizing the importance of optimizing objectives when choosing a particular pooling approach. This highlights the evolving nature of this field of study and the need for careful consideration in pooling layer selection for effective steganalysis.

1. INTRODUCTION

The internet has revolutionized the modern era, fundamentally transforming communication technology. However, publicly available internet connections often lack reliable security, leaving crucial and confidential information vulnerable to theft during data transmission [1]. Protecting digital data is essential, as it is a critical resource in the communication process [2, 3]. This need makes steganography vital for data security [4-6].

Steganography is a technique that embeds private information within digital media without compromising its quality, serving as a crucial tool for protecting digital data [7, 8]. However, steganography has its vulnerabilities, as it can be exploited to spread harmful data, posing risks to users and the digital forensic community [9]. Consequently, detecting and locating modifications in image files used for transmitting confidential information, known as detective [10] and locative [11] steganalysis, become crucial for safeguarding data transmitted over public networks. Steganography and steganalysis are interconnected fields focusing on hidden messages within digital multimedia files [12].

Steganalysis, a classification task in which a robust model determines whether an image file is a cover or stego, involves feature extraction and classification (detecting cover or stego) [9]. This process often utilizes Machine Learning (ML) models such as Support Vector Machines (SVMs) and Ensemble Classifiers (ECs) [13]. However, ML-based steganalysis methods have not fully met broader objectives, prompting researchers to explore deep learning (DL) models for better results in digital image steganalysis [14]. DL, known for its capacity to learn from large datasets, employs Convolutional Neural Networks (CNNs) to extract hidden features from images, enabling more precise detection [15].

While several studies have reviewed state-of-the-art Convolutional Neural Networks (CNNs), a detailed exploration of sensitive features, such as pooling layers and the overall experimental setup, has been lacking. Most have focused solely on increasing classification accuracy, resulting in a general lack of understanding [16]. This research addresses this gap by conducting a sensitivity analysis on an existing CNN model with a pooling layer approach. Pooling layers reduce the dimensionality of input feature maps, thereby lowering computational costs [17]. This research introduces a mixed pooling operation that combines max and average pooling to assess their impact on model performance. The mixed pooling operation assumes an equal contribution from both max and average pooling operations [18]. This paper's sensitivity analysis evaluates how a model's performance depends on its inputs. The study assesses the performance of the works [10, 19-21] presented by employing various combinations of pooling layers. The motivation behind this research stems from insufficient documentation of experimental setups, challenges in replicating CNNs, and inconsistencies in reported outcomes. The contributions of this research are as follows:

(1) *Sensitivity Analysis on CNN Models*: This research performs a detailed sensitivity analysis on an existing CNN model. It introduces a mixed pooling operation that combines max and average pooling to assess their impact on model performance.

(2) *Performance Evaluation with Pooling Layers*: By evaluating the performance of various studies using different combinations of pooling layers, this research addresses vital issues such as insufficient documentation, challenges in replicating CNNs, and inconsistencies in reported outcomes.

(3) *Optimizing Computational Efficiency and Robustness*: The study highlights the role of pooling layers in reducing computational costs and mitigating the effects of noise and minor distortions, emphasizing their importance in enhancing the clarity of desired visual elements.

The subsequent sections of this paper are organized as follows: The "Related Works" section overviews the picture steganalysis framework and discusses related research in the geographical domain. The "Proposed Method" section outlines our methodology. The "Experimental Setup and Results" section details the experimental setup, presents the results, and includes a discussion of the findings. Finally, the "Conclusion" section succinctly wraps up the paper.

2. RELATED WORK

The utilization of Convolutional Neural Networks (CNNs) to identify potentially concealed data within spatial domain images has become commonplace [20-24]. Various studies have explored this avenue, employing fine-tuned CNNs or integrating CNNs with classification algorithms such as fuzzy logic. These models typically incorporate a preprocessing layer to optimize input images, and feature extraction relies on pooling methods to reduce feature dimensionalities, thereby facilitating subsequent extraction and classification. The typical CNN structure consists of three stages: preprocessing, feature extraction, and binary classification. This study focuses on the feature extraction phase, particularly emphasizing the pooling operation. Commonly employed methods include max pooling and average pooling, where max pooling returns the maximum value within a corresponding block, and average pooling computes the mean value [22].

In the model proposed by Ye et al. [20], Spatial Rich Models (SRM) filters are employed in the preprocessing stage, followed by seven convolution layers using the ReLu activation function for feature extraction. The model leverages average pooling, as elucidated in the study, emphasizing its superior performance over max pooling due to considering all values within the pooling region. The phase of features classification, the dense layers, and predictions utilizing the SoftMax layer are utilized.

The Zhu-Net architecture, introduced by Zhang et al. [21], comprises a preprocessing layer, separable convolution layers, convolution layers for feature extraction, a spatial pyramid pooling (SPP) component involving multi-scale operation for average pooling, and a dense layer with SoftMax. While ReLu is employed in the convolution layers, the specific details of the pooling operation are not elaborated.

Enhancing batch normalization, the study [10] adopts average pooling layers with configurations (2,2) for pool size and (2,2) for strides to decrease dimensionality. This architecture integrates four layers of average pooling, demonstrating improved accuracy compared to existing designs, particularly gaining 2.2% and 6% accuracy on S-UNIWARD with 3.4% and 1.7% on WOW. However, the paper lacks clarification on how the varying number of average pooling layers affects accuracy improvement compared to ZHU-Net.

Addressing the sensitivity of steganalysis results to feature map sizes, recent research [9] employs average pooling to reduce feature map sizes, thereby improving robustness to feature position changes and enhancing the network's ability to generalize the feature map. However, this method is considered inferior to 30 SRM filter banks implemented during preprocessing.

In another study [19], the average pooling method is selected to obtain generalized results by combining different values into one average value, aiming to reduce overfitting and enhance accuracy. The results indicate increased accuracy without overfitting, contrasting with a previous study [9] that identified overfitting despite utilizing average pooling to reduce feature map dimensionality.

While numerous studies have offered insights into state-of-the-art CNNs, primarily focusing on enhancing classification accuracy, the current research uniquely contributes by conducting a sensitivity analysis on an existing CNN model, specifically scrutinizing the pooling layer approach. The choice between max pooling and average pooling is pivotal in emphasizing desired visual elements [23, 24]. Building on existing works, this paper systematically assesses how a system's results, particularly a model's performance, depend on its pooling operator choice. The comprehensive examination of critical features such as pooling layers and the overall experimental setup, often overlooked in previous studies, fills a notable gap in the existing literature. This approach ensures a more thorough and nuanced understanding of the dynamics at play in CNN-based steganalysis.

3. METHODOLOGY

3.1 Dataset and experimented CNN architectures

The experiments conducted in this research utilize the Break Our Steganographic System (BOSSBase) database, version 1.01 [25], which contains 10,000 512 × 512-pixel images in Portable Gray Map (PGM) format (8-bit greyscale). The datasets used for model training and testing were first preprocessed, involving resizing the original cover image, creating a stego image using adaptive steganography algorithms, and arranging the data into three sets: training, testing, and validation. This operation ensures a balance between allocating the stego and cover images.

The stego images were generated using two commonly used adaptive steganographic algorithms, Spatial Universal Wavelet Relative Distortion (S-UNIWARD) and Wavelet Obtained Weights (WOW), with a payload capacity of 0.4 Bits Per Pixel (BPP). The resulting dataset consists of 10,000 images each for S-UNIWARD and WOW, respectively. The images are categorized into training, validation, and testing groups to facilitate experiments. Each stego image dataset (WOW and S-UNIWARD) is used on each model. In all experiments, the datasets used include 10,000 images for cover and stego labels each. Based on the schemes in Figures 1-5, each experiment for each model consists of 20,000 datasets that are partitioned into 8000 training sets, 2000 validation sets, and 10,000 testing sets. The percentage division between the cover and the stego is 50%.

Figure 1, illustrating the architecture of a method proposed by Ntivuguruzwa and Ahmad [9], is designed for a steganalysis of images with an initial step involving preprocessing the input image with kernel sizes of 5×5 and 3×3. The activation function used in this preprocessing stage is 3TanH, chosen for its efficiency in enhancing network convergence. In the feature extraction stage, a depthwise separable 2D convolution is implemented, consisting of four layers combined with 2D convolution operations. The LeakyReLU activation function is employed within this architecture to prevent gradient loss and optimize learning efficiency. Incorporating Batch Normalization during the training phase is a strategic choice to prevent the loss of gradients, thereby avoiding network overloading and enhancing the learning rate, which expedites network convergence.

In Figure 2, another architecture consists of a preprocessing phase involving the convolution of 30 Spatial Rich Models (SRM) filters, each with a size of 5×5. Notably, this layer is non-trainable, ensuring the convolution remains unchanged throughout the training stage. The convolutional layers in this phase use padding set to 'same,' 30 filters, 1×1 strides, and the 3TanH activation function. Transitioning to the feature extraction stage, multiple layers are incorporated, including depthwise convolutions, separable convolutions, and traditional convolutional layers. After Batch Normalization, pooling layers are implemented to reduce dimensionality. This approach combines Average Pooling, Max Pooling, and Mixed Pooling with a fixed pool size of 2×2 and strides of 2×2. The Exponential Linear Unit (ELU) is the activation function for all convolutions, including separable ones. Concluding the feature extraction stage, global average pooling is applied to prepare the features for classification [10].

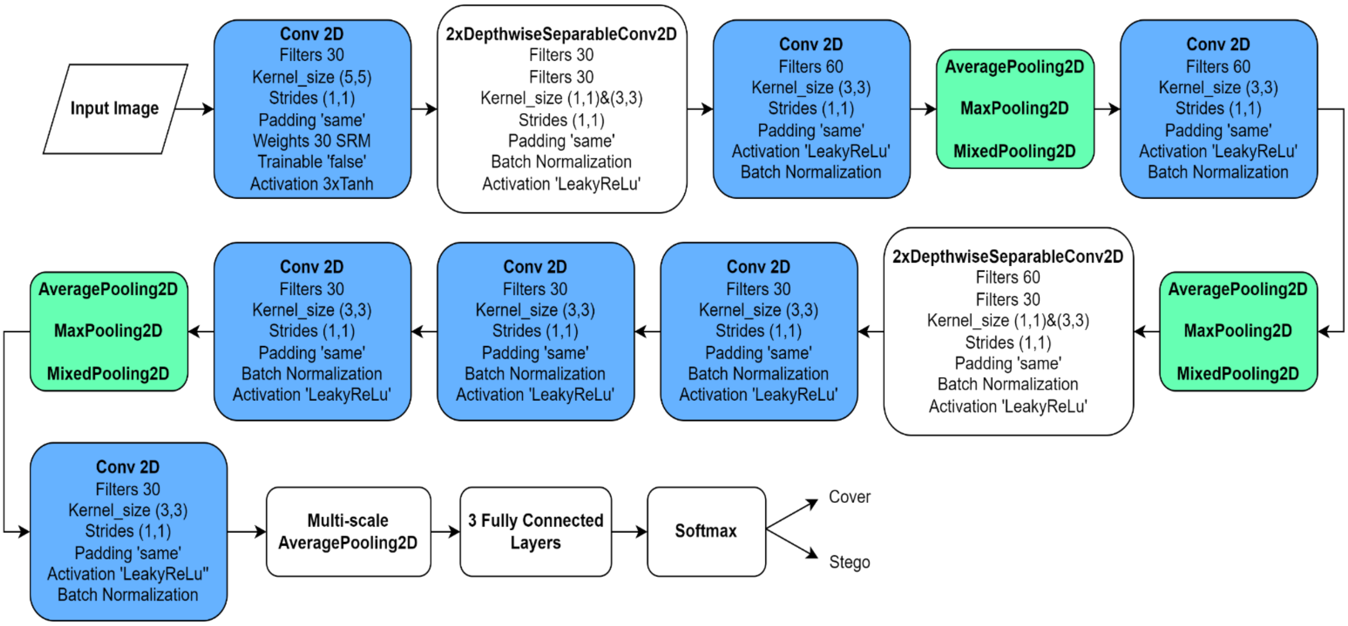


Figure 1. The CNN architecture [9]

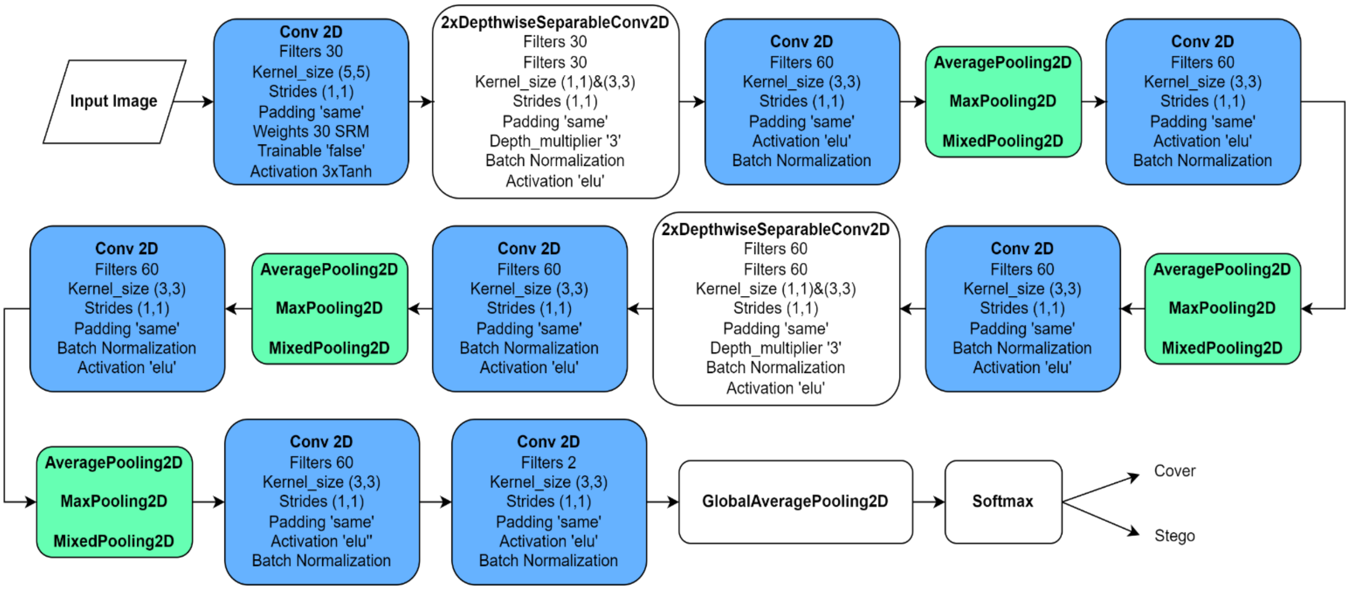


Figure 2. The CNN architecture [10]

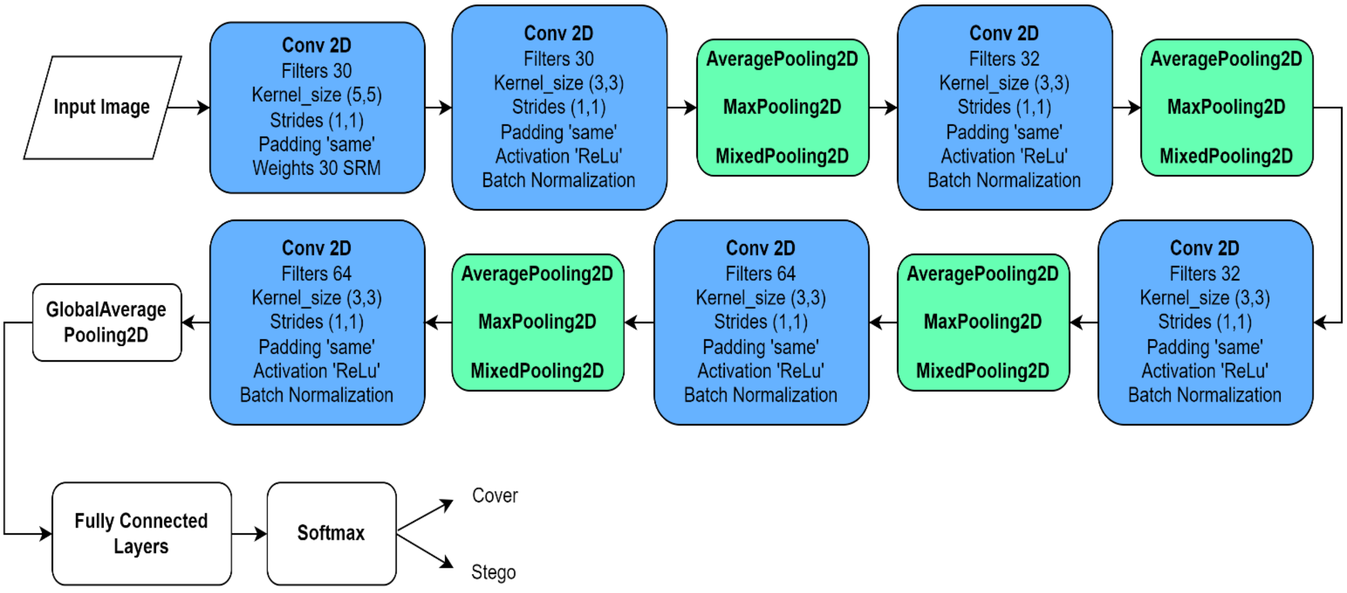


Figure 3. The CNN architecture [19]

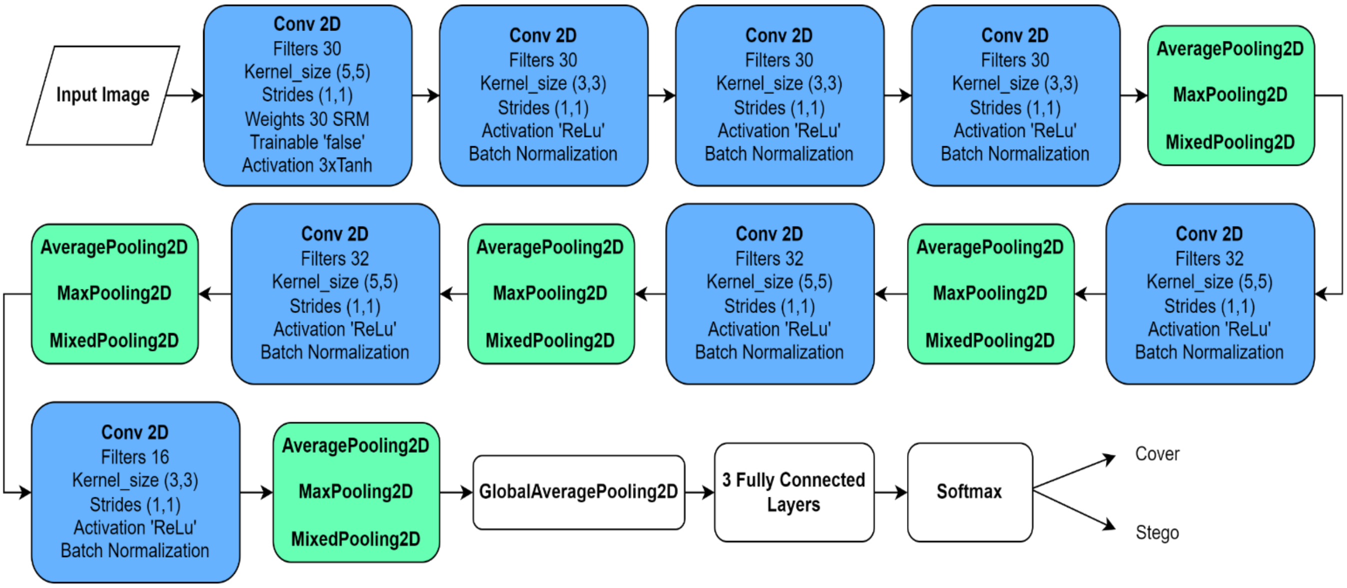


Figure 4. The CNN architecture [20]

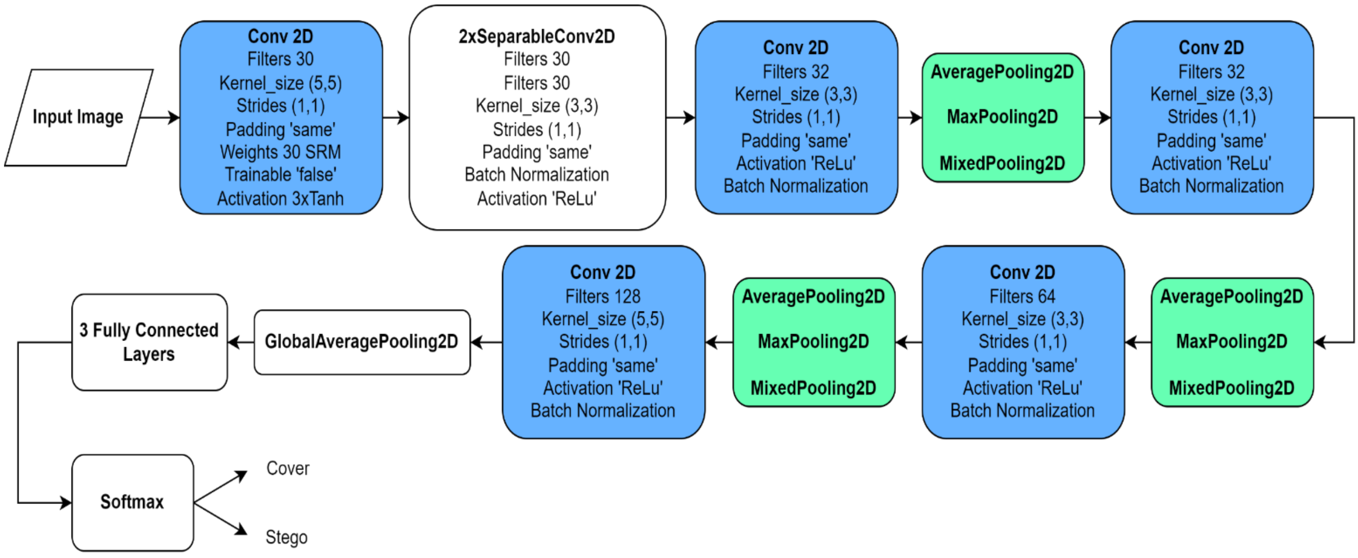


Figure 5. The CNN architecture [21]

In the model presented in the study by de La Croix and Ahmad [19], the initial input consists of a 256×256 sized digital image. To enhance network convergence, the preprocessing stage incorporates Spatial Rich Models (SRM) with a 3Tanh activation function. The subsequent feature extraction phase comprises five convolution layers, fostering the extraction of hierarchical features. Four pooling layers are strategically placed to reduce dimensionality and capture essential information. The Rectifying Linear Unit (ReLu) activation function introduces non-linearity, contributing to the model's capacity for learning intricate patterns. Concluding the architecture, global average pooling is applied to consolidate abstracted features. This is followed by fully connected layers that facilitate the amalgamation of high-level features for adequate classification. The final classification stage uses the SoftMax function to produce probabilistic predictions. The architecture, visually represented in Figure 3, underscores a comprehensive design strategy, leveraging SRM in preprocessing, employing convolution and pooling layers for feature extraction, and culminating in global average pooling and fully connected layers for robust classification.

The architecture introduced by Ye et al. [20] and illustrated in Figure 4 is characterized by a well-defined structure designed for effective steganalysis. The initial preprocessing stage involves a single convolution layer, employing a kernel size of (5,5). It incorporates a 30 Spatial Rich Models (SRM) filter. This setup aids in enhancing the network's convergence capabilities. The subsequent feature extraction phase encompasses seven 2D convolution layers. The kernel size is set to (3,3) for the initial and final layers.

In contrast, a (5,5) kernel size is applied to the intermediate layers. The Rectifying Linear Unit (ReLu) activation function is utilized throughout this stage to introduce non-linearity, enabling the model to capture intricate patterns effectively. Including Batch Normalization during training is a strategic decision to prevent gradient loss. This helps to avoid network overload, improves the learning rate, and accelerates network convergence.

Pooling layers play a crucial role in reducing dimensionality and extracting essential features. The architecture incorporates a combination of pooling approaches, including average pooling, max pooling, and mixed pooling. This diversity in pooling methods enhances the model's adaptability to different data types, contributing to its robust performance. Notably, the proposed model, depicted in Figure 4, does not utilize separable convolutions, opting for a design that emphasizes simplicity and efficiency. Global average pooling is employed for the subsequent classification phase, followed by three fully connected layers. The SoftMax activation function is applied to generate predictions, ensuring a probabilistic output for adequate classification. The overall architecture underscores a thoughtful integration of convolutional and pooling layers, emphasizing versatility and efficiency in steganalysis.

The architecture proposed by Zhang et al. [21] exhibits a well-defined structure tailored for efficient steganalysis. The initial stage involves preprocessing a 256×256 digital image, using 3Tanh activation functions and 30 Spatial Rich Models (SRM) filters. This preprocessing step is crucial for enhancing network convergence and preparing the input for subsequent feature extraction. The feature extraction stage incorporates a thoughtful design, including a separable convolution layer at the outset. This design choice aims to capture essential features effectively while optimizing computational efficiency. Four 2D convolution layers are employed, each utilizing the Rectifying Linear Unit (ReLu) activation function. The use of ReLu contributes to the introduction of non-linearity, enabling the model to learn complex patterns inherent in stego and cover images. A notable feature of this architecture is its three-layer pooling approach in the feature extraction stage. Pooling layers are pivotal in reducing dimensionality and enhancing the model's ability to discern relevant features. The choice of a three-layer pooling strategy underscores a nuanced approach to feature extraction, ensuring that essential information is retained. Similar to the previously discussed model, a familiar pattern is observed in the classification stage. Global average pooling is employed, concisely representing the extracted features. Three fully connected layers follow, incorporating the SoftMax activation function for accurate and probabilistic predictions. The overall architecture, illustrated in Figure 5, reflects a balanced integration of preprocessing, feature extraction, and classification components, emphasizing efficiency and effectiveness in steganalysis.

3.2 Method of sensitivity testing through maximum, average, and mixed pooling operations

This research focuses on the sensitivity of CNNs to these different pooling layers to understand their impact on model performance and robustness. By exploring the effectiveness of maximum, average, and mixed pooling, the study aims to identify optimal down-sampling techniques that enhance feature extraction while mitigating computational costs and overfitting. The rationale for choosing maximum, average, and mixed pooling layers in this research is grounded in their distinct and complementary characteristics:

(1) *Maximum Pooling:* This method selects the maximum value within each pooling window. It is highly effective in preserving the most prominent features of an input image, which can be critical for tasks that require capturing robust, decisive features. Max pooling is widely recognized for its ability to maintain feature invariance and reduce spatial dimensions, making it a standard choice in many CNN architectures.

(2) *Average Pooling*: Unlike max pooling, average pooling computes the average of all values within each pooling window. This method smooths out feature maps, retaining the overall structure while reducing the impact of noise and extreme values. It is beneficial for capturing the background information and general trends within the input data, providing a balanced feature map representation.

(3) *Mixed Pooling*: Mixed pooling combines the strengths of both max and average pooling by averaging their outputs. This hybrid approach aims to leverage the advantages of both methods, preserving prominent features through max-pooling while maintaining the overall structural integrity with average pooling. Mixed pooling can provide a more nuanced and robust feature representation, which can be beneficial in scenarios where both sharp features and general trends are essential.

A. Method with the maximum pooling layer

Based on Figure 6, the maximum activation value in the pooling area is used to choose activation values in max pooling. To mathematically express the maximum pooling operation, we refer to the Eq. (1) with x, the vector of activation values in the input image's pooling region.

$$f_{max}(x) = max(x_n) \qquad (1)$$

It is essential to highlight that the main drawback of max pooling is its exclusive consideration of the maximum element within the pooling region, disregarding all other components. This limitation can lead to information loss, mainly when discriminative features are present in elements with high activation values that are not accounted for in the pooling process.

B. Method with the average pooling layer

As a method for selecting activation values, Figure 7 shows the functionality of the average pooling operation, which relies on computing the average activation within the designated region. The process involves determining the average activation value of the specified region to facilitate pooling as mathematically expressed in Eq. (2), considering x as the vector of activation values in the input image's pooled region. However, a notable challenge emerges when a majority of the activation values in the region are zero. In such instances, the average is significantly diminished, resulting in pooled features with values approaching zero or precisely zero. Consequently, during subsequent processing stages, the network may encounter difficulties recognizing and identifying dominant features due to this potential loss of information. This emphasizes the critical need to address the impact of zero or near-zero values on the effectiveness of average pooling in feature extraction.

$$f_{avg}(x) = \frac{1}{N}\sum_{n=1}^{N} x_n \qquad (2)$$

C. Method with the mixed pooling layer

The combination of average and maximum pooling, termed mixed pooling, involves adopting a comprehensive approach that harnesses the combined strengths of both average and max pooling operations to enhance overall network performance. As illustrated in Figure 8, the mixed pooling operation is mathematically expressed in (3), considering x as a scalar factor known as mixing proportion and $\alpha_l \in [0,1]$. An illustration of mixed pooling is shown in Figure 8. While both average and max pooling exhibit effectiveness in specific data scenarios, determining the superior approach for addressing novel challenges remains uncertain. The inherent dynamism of natural images underscores the potential drawbacks of max pooling and average pooling, which could impede their optimal utilization in Convolutional Neural Networks (CNNs). Consequently, this study introduces a comparative approach for the maximum, average, and mixed pooling to each architecture as a comparative benchmark, providing insights into the varied utilization of pooling operations in CNNs.

$$f_{mix}(x) = \alpha_l \cdot f_{max}(x) + (1 - \alpha_l) \cdot f_{avg}(x) \qquad (3)$$

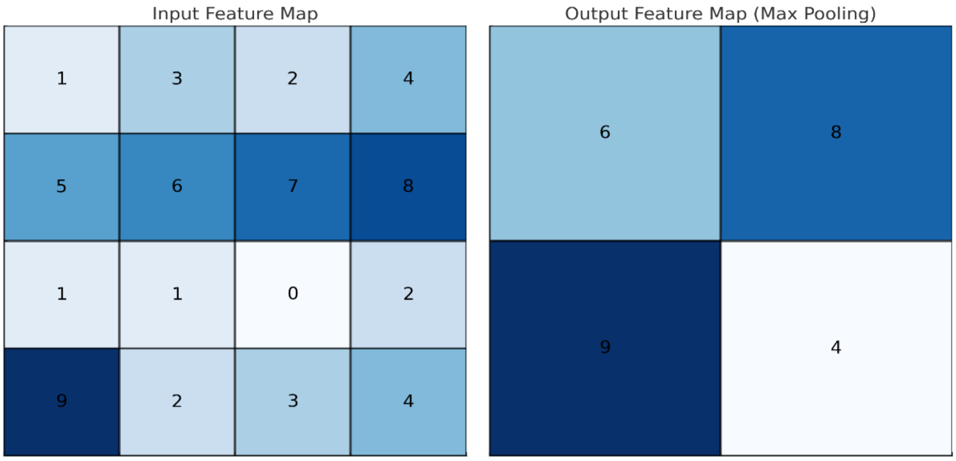


Figure 6. Illustration of the proposed maximum pooling operation

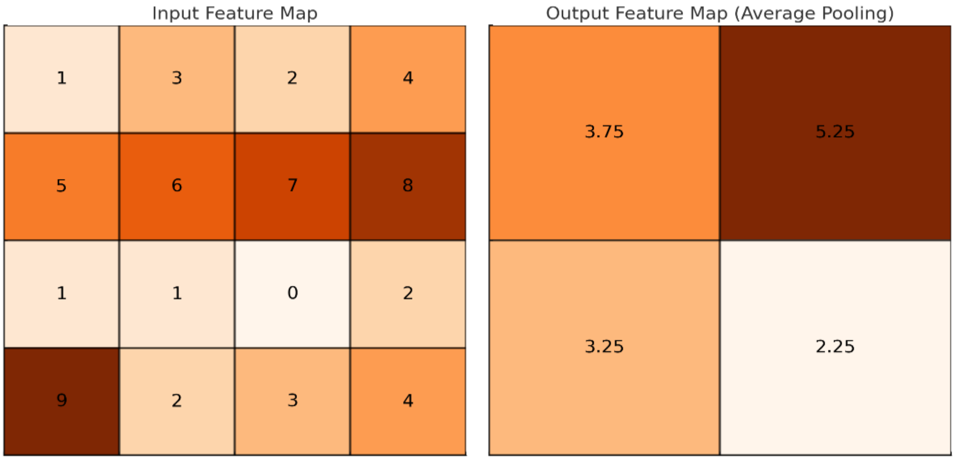


Figure 7. Illustration of the proposed average pooling operation

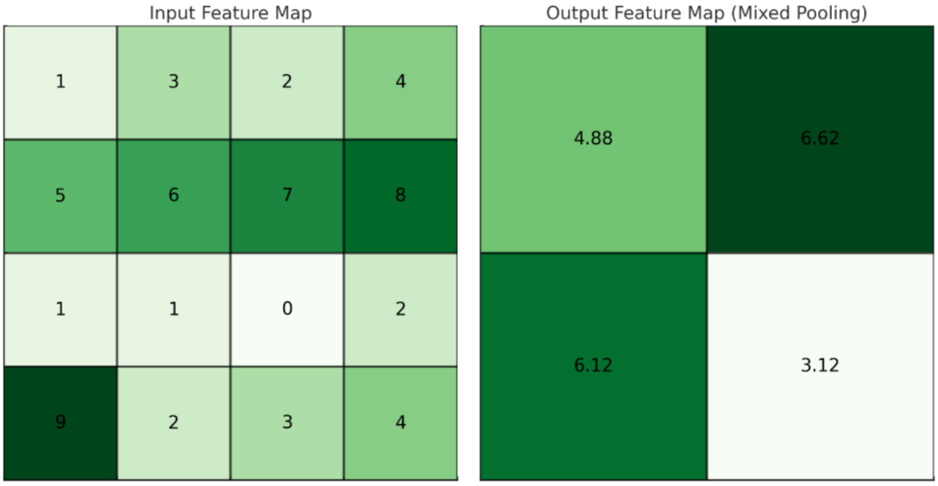


Figure 8. Illustration of the proposed mixed pooling operation

3.3 Hyper-parameters and training

In the training approach, hyperparameters and optimization are strategically selected to enhance the model's learning process and accelerate the model's convergence with improved performance. 2D convolutions and deep separable layers are combined with the Glorot kernel to prevent vanishing or exploding gradients and keep the learning process stable. The BN layer configuration is done with these specific settings: 0.2 for momentum, 0.002 for an epsilon, and 0.4 for a renorm momentum. These parameters are set to adapt smoothly to the data allocations with reduced overfitting that may happen in the training process. The proposed model uses an Adam optimizer to handle any irregularities that may arrive at the gradient variations, and the model learnability rate is set to 0.001 for the sake of balanced accuracy and model convergence.

The size for the training batch remains constant at 32 frames across all architectures, fostering stability and uniformity throughout the training process. Convergence is achieved within 100 training epochs, demonstrating the efficiency of the chosen hyperparameters. Key parameters, including renorm momentum (0.4), epsilon (0.001), and batch normalization momentum (0.2), contribute to the adaptability and performance of the models. Initialized with a Glorot uniform kernel, convolutional layers ensure consistent feature extraction across architectures.

The Adam optimizer is employed for the works in [9, 10, 19], with a learning rate of 0.001, beta 1 set to 0.9, beta 2 to 0.999, and epsilon at 1e-08. For the works in some studies [20, 21], optimization algorithms are based on reported parameters, utilizing Stochastic Gradient Descent (SGD) and RMSprop. In the final stage of the architectures, predictions directly employ a SoftMax activation function. Binary cross-entropy loss is selected for the dual-class classification of cover and stego.

This standardized training approach ensures a consistent and effective model development, emphasizing stability, efficiency, and adaptability across diverse architectures. The selected hyperparameters and optimization strategies underscore a thoughtful consideration of the models' intricacies and objectives.

4. EXPERIMENT AND RESULT

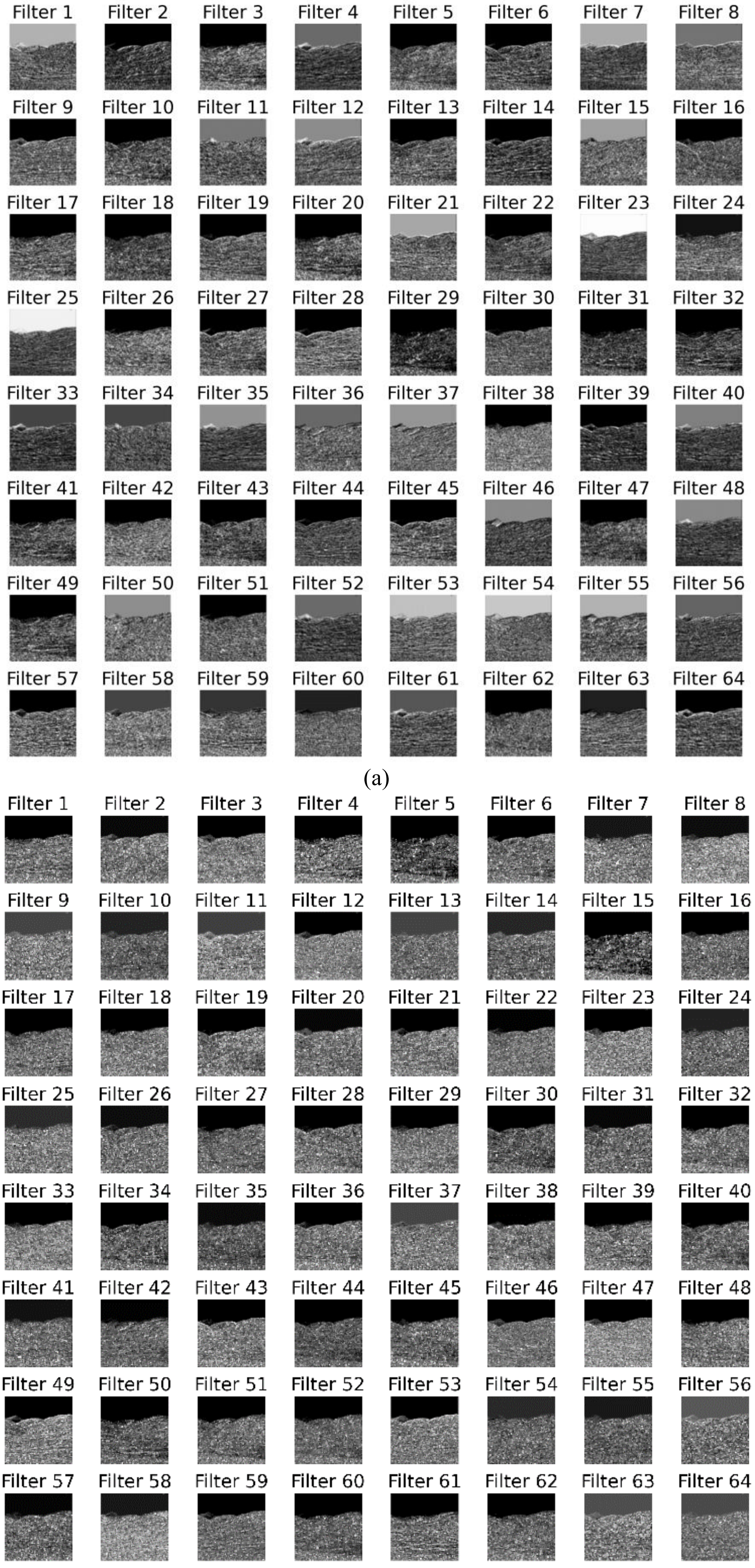
Based on Figure 9, the feature map generated by average pooling tends to have a smoother gradation as the feature map values are taken from the average result, but it may cause the loss of some essential details. On the other hand, the feature map generated by max pooling has a very high contrast, as only the maximum value is retained while other values are ignored. Mixed pooling tries to get the best advantage of both types of pooling by producing a feature map that retains essential details while detecting solid features. The results tend to be balanced and stable, with areas that have smooth transitions and areas that show prominent features. This pooling process may increase computational complexity and time but can result in a more informative and robust feature map.

Table 1. Accuracy results for each pooling operation on considered model architectures

Model Architecture	Pooling	S-UNIWARD 0.4 BPP	WOW 0.4 BPP
Architecture in the study [9]	Average	84	90
	Max	81	85
	Mixed	81	84
Architecture in the study [10]	Average	89	91
	Max	80	83
	Mixed	82	84
Architecture in the study [19]	Average	83	87
	Max	80	84
	Mixed	79	83
Architecture in the study [20]	Average	68	76
	Max	73	82
	Mixed	82	82
Architecture in the study [21]	Average	84	88
	Max	86	85
	Mixed	84	86

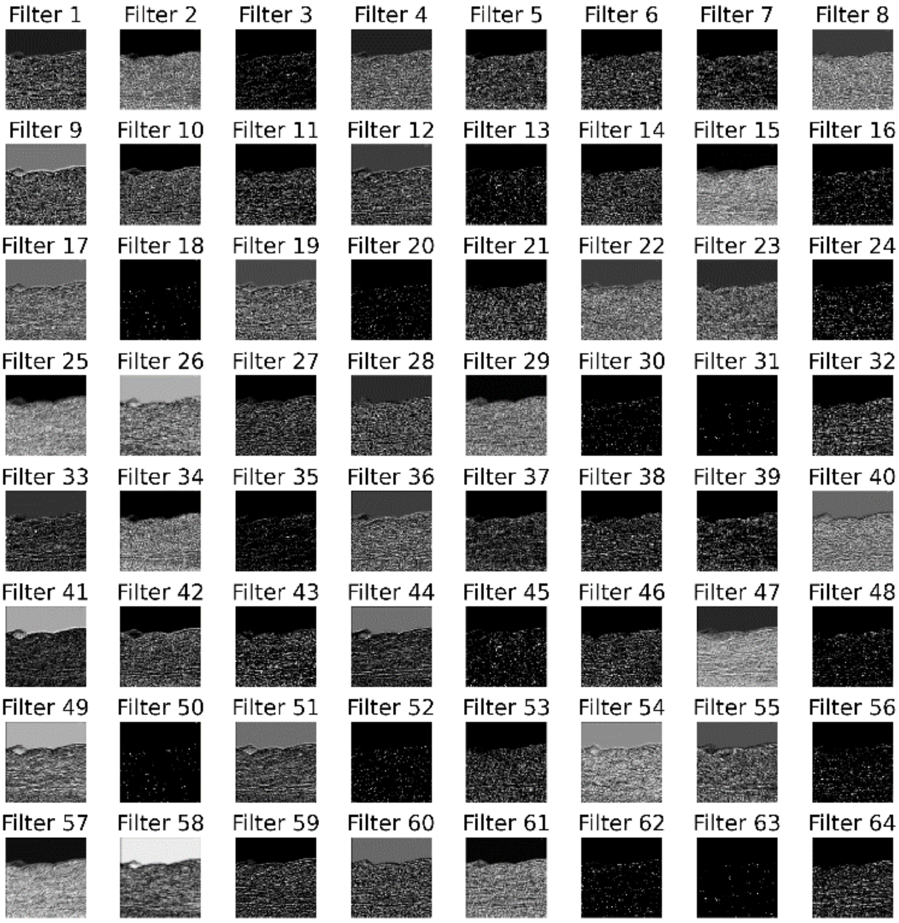
The comprehensive analysis in Table 1 thoroughly examines the results derived from diverse pooling layers applied to five distinct architectures within the BOSSBase 1.01 database. Utilizing the S-UNIWARD and WOW steganography methods, each featuring a 0.4 BPP payload, underscores distinctive trends in the reported findings. Notably, average pooling demonstrates superiority across three architectures: Ntivuguruzwa and Ahmad [9] achieves a commendable 84% and an impressive 90%, the study of de La Croix and Ahmad [19] attains a substantial 83%. A noteworthy 87%, and Reinel et al. [10] record a remarkable 89% and a striking 91% for S-UNIWARD and WOW, respectively.

Adding intrigue to the insights, the CNN model detailed in the study of Zhang et al. [21] reaches the pinnacle of accuracy, securing results of 86% (with max pooling) and 88% (with average pooling) for the 0.4 BPP payload. In contrast, the distinctive approach of mixed pooling in the study of Ye et al. [20] yields the highest accuracy, registering an appreciable 82% for both S-UNIWARD and WOW payloads.



(a)

(b)



(c)

Figure 9. Feature map visualization of (a) average pooling, (b) max pooling, (c) mixed pooling

Table 2. Recall, precision, and f1-score results for each pooling with the WOW steganography method (0.4 BPP)

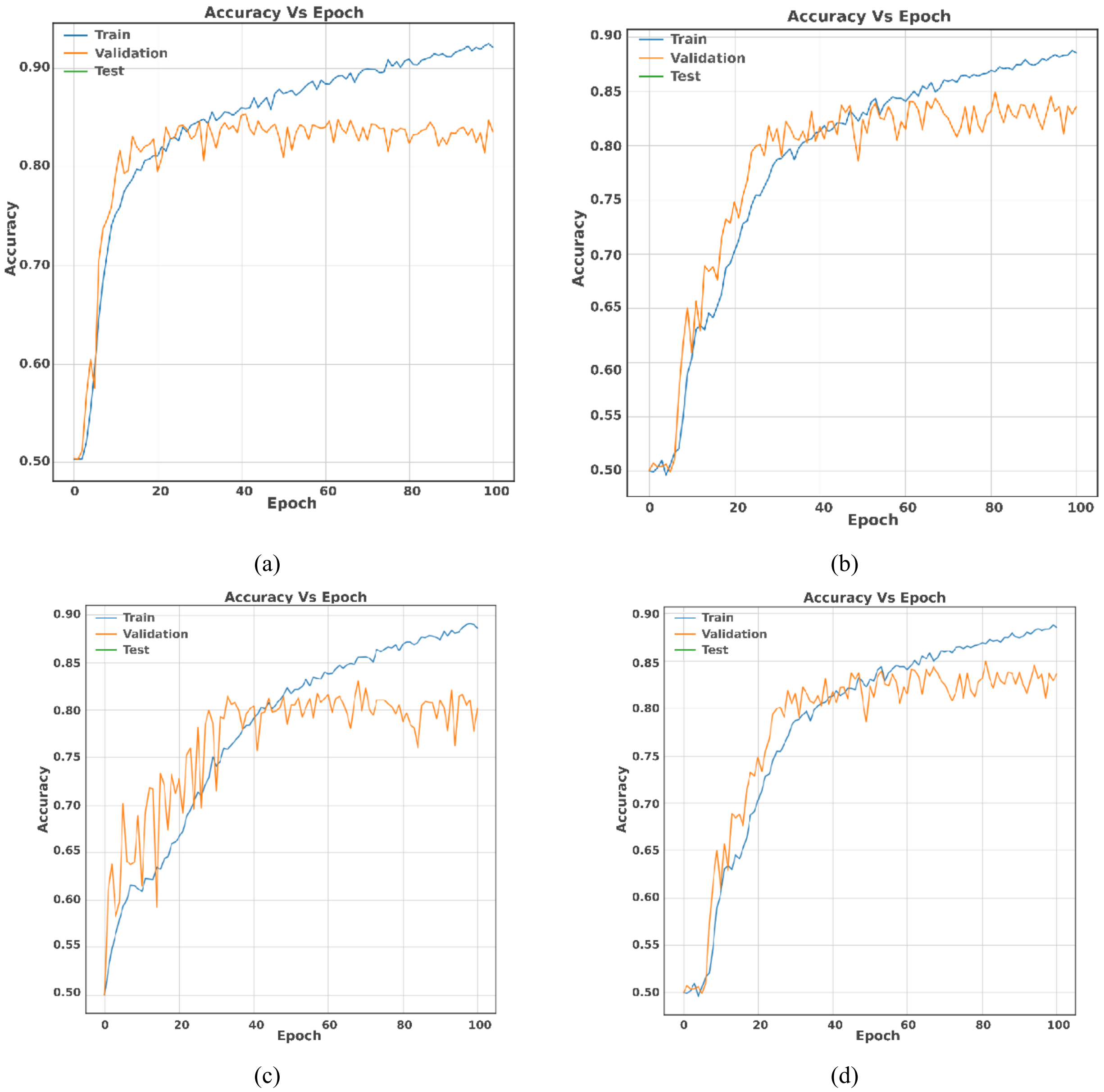
Model Architecture	Pooling	WOW (0.4 BPP)		
		Recall	Precision	F1-Score
Architecture in the study [9]	Average	84	85	84
	Max	84	84	84
	Mixed	85	85	85
Architecture in the study [10]	Average	83	83	83
	Max	85	85	85
	Mixed	84	85	84
Architecture in the study [19]	Average	85	85	85
	Max	83	83	83
	Mixed	85	85	85
Architecture in the study [20]	Average	85	85	85
	Max	83	83	83
	Mixed	85	85	85
Architecture in the study [21]	Average	85	85	85
	Max	87	88	87
	Mixed	86	86	86

A nuanced examination of activation functions reveals their nuanced impact on accuracy. Notably, the eLu activation in the study of Reinel et al. [10] and LeakyReLu in the study by Ntivuguruzwa and Ahmad [9] showcase superior accuracy, reaching an impressive 90% with the application of average pooling. Moreover, the pivotal role of optimization algorithms comes to the forefront, with SGD in the study by Ye et al. [20] proving more effective when paired with mixed pooling, outperforming the Adam algorithm.

Delving into the accuracy curves provides valuable insights into overfitting concerns. Figures 10, 11, and 12 meticulously demonstrate the efficacy of mixed pooling in mitigating the observed overfitting challenges associated with max pooling. However, the scenario in the study [20] offers a unique perspective; although max pooling does not induce overfitting, Figure 13 showcases that mixed pooling still contributes to superior accuracy.

Further analysis was conducted, based on Table 2; average pooling performs well in most models, such as the ICTAS and Ye-Net models, resulting in 85% for each evaluation metric. However, in some cases, such as the Zhu-Net and GBRAS-Net models, average pooling shows lower results, up to 83%, than max and mixed pooling. For max pooling, it shows varying performance. In the Zhu-Net model, max pooling reaches 88% and is the highest for each model. However, its performance decreased in the ICTAS and Ye-Net models to 83% for each metric evaluation. Mixed pooling consistently provided the best or most stable results in all models, reaching 86% in the Zhu-Net model. This suggests that combining average and max pooling can capture more relevant information and thus improve overall performance.

The strategic role of pooling operations in steganalysis becomes apparent, with average pooling emerging as the go-to choice for its accuracy in preserving steganographic noise. Nonetheless, max pooling, under specific conditions such as image characteristic preservation, is an effective strategy for optimizing accuracy. Adopting mixed pooling presents a promising avenue, leveraging a balanced combination of average and max pooling to address overfitting concerns, thereby enhancing the robustness of steganalysis models.



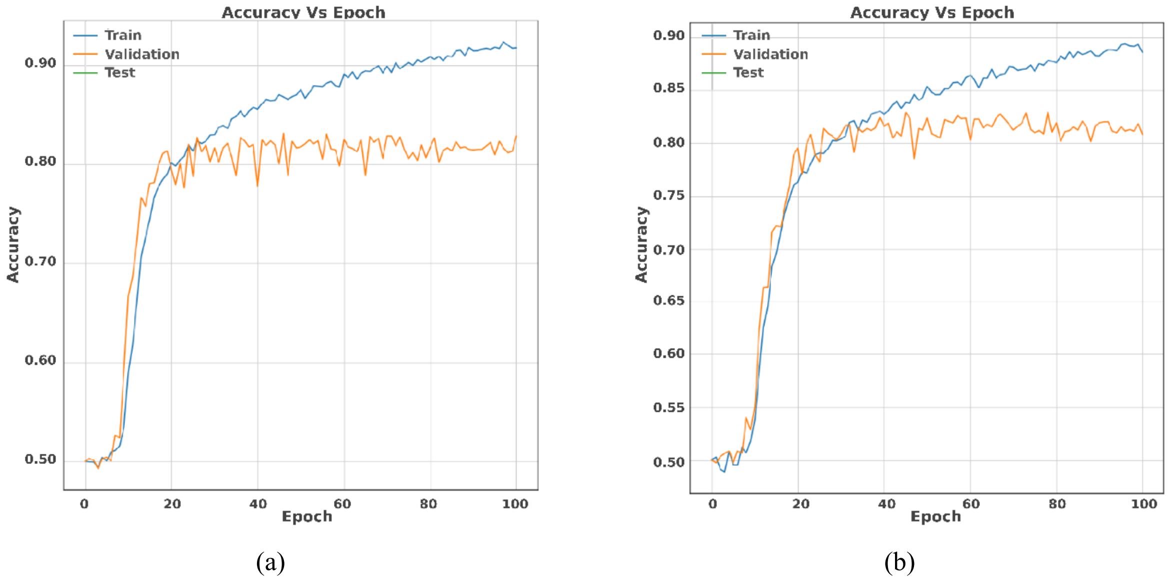
(a)

(b)

(c)

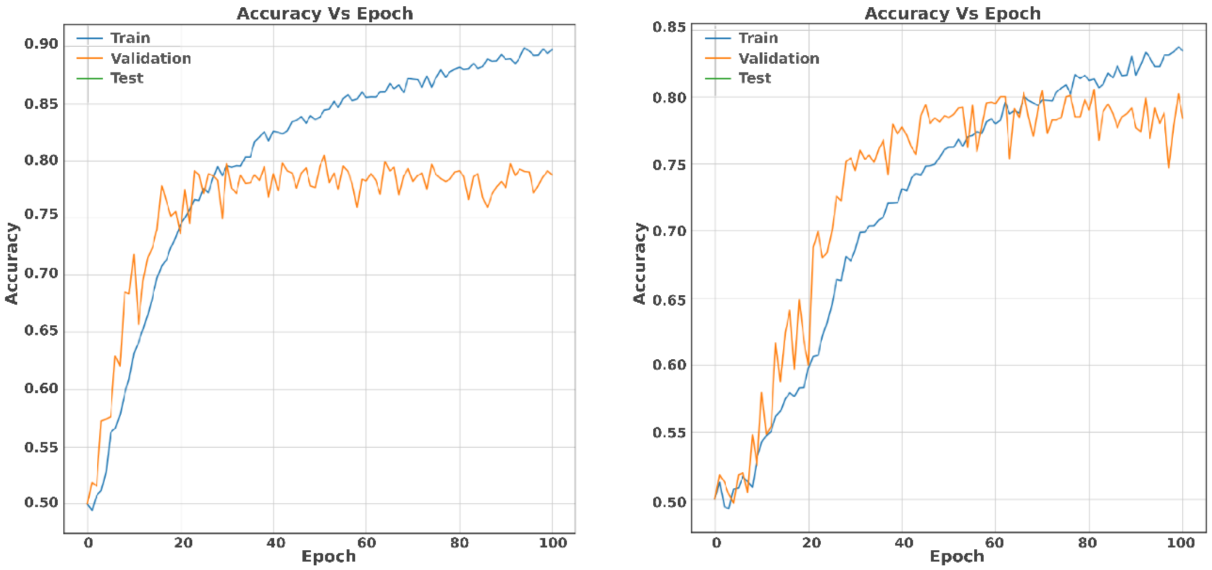
(d)

Figure 10. Training and validation accuracy curves for the method in the study [9] with 0.4 BPP. a) WOW with max pooling b) WOW with mixed pooling c) S-UNIWARD with max pooling d) S-UNIWARD with mixed pooling



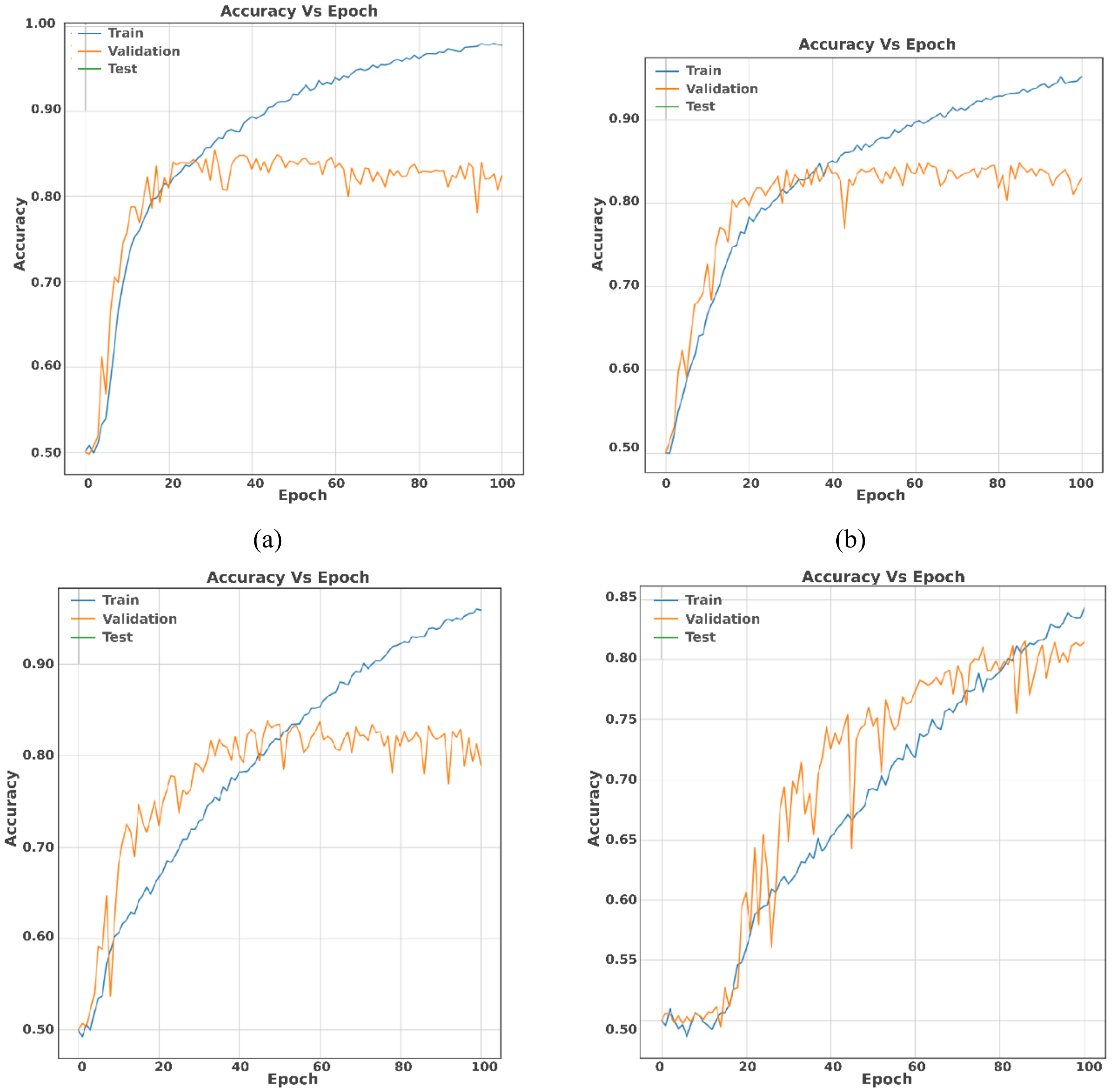
(a)

(b)



(c) (d)

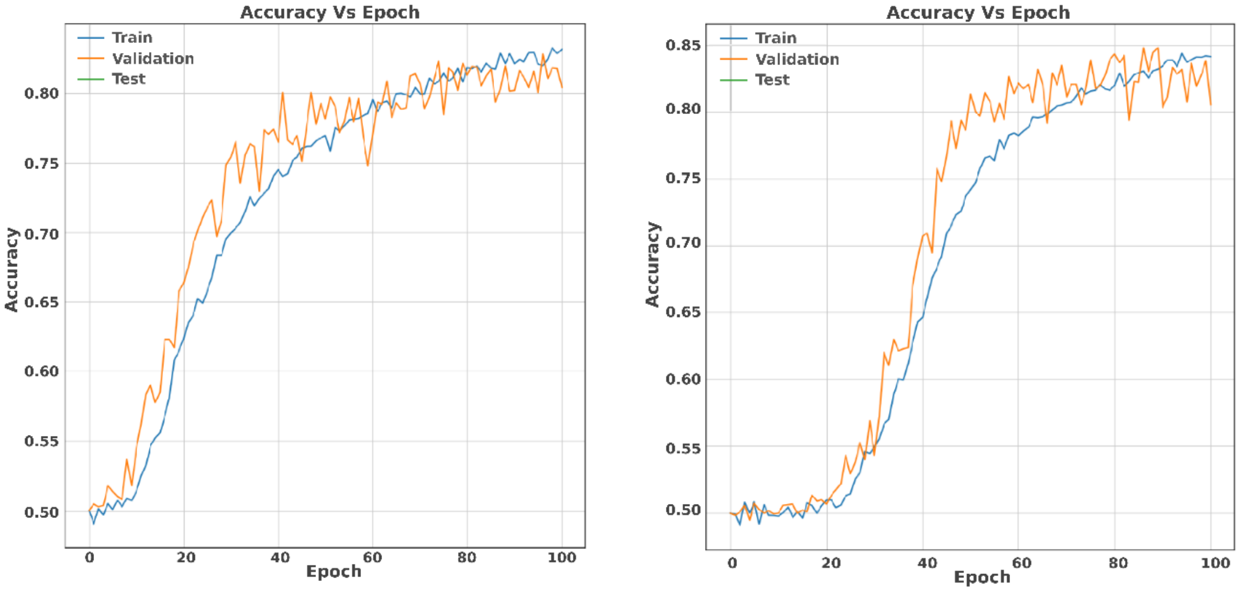
Figure 11. Training and validation accuracy curves for the method in the study [19] with 0.4 BPP. a) WOW with max pooling b) WOW with mixed pooling c) S-UNIWARD with max pooling d) S-UNIWARD with mixed pooling



(a) (b)

(c) (d)

Figure 12. Training and validation accuracy curves for the method in the study [10] with 0.4 BPP. a) WOW with max pooling b) WOW with mixed pooling c) S-UNIWARD with max pooling d) S-UNIWARD with mixed pooling



(a)

(b)

Figure 13. Training and validation accuracy curves for the method in the study [20] with 0.4 BPP. a) WOW with max pooling b) WOW with mixed pooling

5. CONCLUSIONS

The experiments have yielded a spectrum of results, underscoring the critical need for meticulous documentation in this evolving research domain. Leveraging the widely utilized BOSSBase 1.01 dataset with PGM format (8-bit greyscale), encompassing 10,000 datasets at a resolution of 256x256 pixels, this research meticulously dissects the dataset into 4000 training, 1000 validation, and 5000 testing instances. The investigated CNN architectures include the ones previously reported in some studies [9, 10, 19-21], employing a pooling layer approach and scrutinizing the comparative performance of average pooling, max pooling, and mixed pooling. Accuracy analysis reveals that for architectures [9, 10, 19], average pooling emerges as the optimal choice, consistently yielding superior accuracy. The nuanced interaction of pooling operations with other hyper-parameters, exemplified by the SGD optimization algorithm in the model referenced in the study by Ye et al. [20], accentuates the need for meticulous customization to achieve optimal accuracy. A common thread emerges in the accuracy curves: the pooling layer's pivotal role in curbing overfitting across diverse architectures. It is imperative to recognize that pooling operations play nuanced roles, requiring careful consideration in model development. The interplay of various layers and hyper-parameters significantly influences accuracy outcomes, with this paper shedding light on the sensitivity of CNNs to the pooling layer, offering a foundational understanding for future research in this dynamic field.

In the realm of future research, several promising avenues emerge from the findings of this study. Firstly, exploring novel pooling techniques beyond the conventional average, max, and mixed pooling could unveil additional insights into optimizing steganalysis performance. Investigating the interplay between pooling operations and activation functions, especially in diverse architectures, presents an intriguing direction for enhancing model robustness. Additionally, delving deeper into the impact of pooling layers on specific steganographic methods, such as S-UNIWARD and WOW, could yield tailored insights for refining steganalysis under different scenarios. Lastly, considering the temporal dimension by incorporating sequential data or exploring recurrent neural networks may offer a more comprehensive understanding of steganalysis dynamics.

ACKNOWLEDGMENT

The authors gratefully thank the collaborated members of the NCC laboratory and all the research group members who invaluably contributed to this work in one way or another.

REFERENCES

[1] Setiadi, D.R.I.M. (2022). Improved payload capacity in LSB image steganography uses dilated hybrid edge detection. Computer and Information Sciences, 34(2): 104-114. https://doi.org/10.1016/j.jksuci.2019.12.007

[2] Narayana, V.L., Kumar, N.A. (2018). Different techniques for hiding the text information using text steganography techniques: A survey. Ingénierie des Systèmes d'Information, 23(6): 115-125. https://doi.org/10.3166/isi.23.6.115-125

[3] Hu, K., Wang, M., Ma, X., Chen, J., Wang, X., Wang, X. (2024). Learning-based image steganography and watermarking: A survey. Expert Systems with Applications, 249: 123715. https://doi.org/10.1016/j.eswa.2024.123715

[4] Sarmah, D.K., Kulkarni, A.J. (2019). Improved cohort intelligence high capacity, swift, and secure approach to JPEG image steganography. Journal of Information Security and Applications, 45: 90-106. https://doi.org/10.1016/j.jisa.2019.01.002

[5] Hassan, F.S., Gutub, A. (2022). Improving data hiding within colour images using hue component of HSV

colour space. CAAI Transactions on Intelligence Technology, 7(1): 56-68. https://doi.org/10.1049/cit2.12053

[6] Pei, Y., Luo, X., Zhang, Y., Zhu, L. (2020). Multiple images steganography of JPEG images based on optimal payload distribution. Computer Modeling in Engineering & Sciences, 125(1): 417-436. https://doi.org/10.32604/cmes.2020.010636

[7] Yang, H., Xu, Y., Liu, X., Ma, X. (2024). PRIS: Practical robust invertible network for image steganography. Engineering Applications of Artificial Intelligence, 133: 108419. https://doi.org/10.1016/j.engappai.2024.108419

[8] 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

[9] 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

[10] Reinel, T.S., Brayan, A.A.H., Alejandro, B.O.M., Alejandro, M.R., Daniel, A.G., Alejandro, A.G.J., 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

[11] Qiao, T., Luo, X., Pan, B., Chen, Y., Wu, X. (2022). Toward steganographic payload location via neighboring weight algorithm. Security and Communication Networks, 2022(1): 1400708. https://doi.org/10.1155/2022/1400708

[12] Yang, H., He, H., Zhang, W., Cao, X. (2020). Fedsteg: A federated transfer learning framework for secure image steganalysis. IEEE Transactions on Network Science and Engineering, 8(2): 1084-1094. https://doi.org/10.1109/TNSE.2020.2996612

[13] Maddumala, V.R. (2020). Using a convolutional neural network, a weight-based feature extraction model on multifaceted multimedia big data. Ingénierie des Systèmes d'Information, 25(6): 729-735. https://doi.org/10.18280/isi.250603

[14] Talai, Z., Kherici, N., Bahi, H. (2023). Comparative study of CNN structures for Arabic speech recognition. Ingénierie des Systèmes d'Information, 28(2): 327-333. https://doi.org/10.18280/isi.280208

[15] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8: 1-74. https://doi.org/10.1186/s40537-021-00444-8

[16] Challa, R., Rao, K.S. (2021). A hybrid approach is used to detect objects from images using a fisher vector and PSO-based CNN. Ingénierie des Systèmes d'Information, 26(5): 483-489. https://doi.org/10.18280/isi.260508

[17] Skourt, B.A., El Hassani, A., Majda, A. (2022). Mixed-pooling-dropout for convolutional neural network regularization. Journal of King Saud University-Computer and Information Sciences, 34(8): 4756-4762. https://doi.org/10.1016/j.jksuci.2021.05.001

[18] Laxminarayanamma, K., Krishnaiah, R.V., Sammulal, P. (2022). Enhanced CNN model for pancreatic ductal adenocarcinoma classification based on proteomic data. Ingénierie des Systèmes d'Information, 27(1): 127. https://doi.org/10.18280/isi.270115

[19] de La Croix, N.J., Ahmad, T. (2023). Toward hidden data detection via local features optimization in spatial domain images. In 2023 Conference on Information Communications Technology and Society (ICTAS), Durban, South Africa, pp. 1-6. https://doi.org/10.1109/ICTAS56421.2023.10082736

[20] 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

[21] 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.

[22] Yu, D., Wang, H., Chen, P., Wei, Z. (2014). Mixed pooling for convolutional neural networks. In Rough Sets and Knowledge Technology: In 9th International Conference, RSKT 2014, Shanghai, China, pp. 364-375. https://doi.org/10.1007/978-3-319-11740-9_34

[23] De La Croix, N.J., Ahmad, T., Han, F. (2023). Enhancing secret data detection using convolutional neural networks with fuzzy edge detection. IEEE Access, 11: 131001-131016. https://doi.org/10.1109/ACCESS.2023.3334650

[24] Sharma, T., Verma, N.K., Masood, S. (2023). Mixed fuzzy pooling in convolutional neural networks for image classification. Multimedia Tools and Applications, 82(6): 8405-8421. https://doi.org/10.1007/s11042-022-13553-0

[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