

Preprocessing Strategy to Improve the Performance of Convolutional Neural Networks Applied to Steganalysis in the Spatial Domain

Mario Alejandro Bravo-Ortiz^{1,*}, Esteban Mercado-Ruiz¹, Juan Pablo Villa-Pulgarin¹,
Harold Brayan Arteaga-Arteaga¹, Gustavo Isaza², Raúl Ramos-Pollán³,
Manuel Alejandro Tamayo-Monsalve^{1,4}, and Reinel Tabares-Soto^{1,2}

¹Departamento de Electrónica y Automatización, Universidad Autónoma de Manizales, Manizales, Caldas, Colombia

²Departamento de Sistemas e Informática, Universidad de Caldas, Manizales, Caldas, Colombia

³Departamento de Ingeniería de Sistemas, Universidad de Antioquia, Medellín, Antioquia, Colombia

⁴Facultad de Ciencias e Ingeniería, Universidad de Manizales, Manizales, Caldas, Colombia

Email: mario.bravoo@autonoma.edu.co (M.A.B.O.)

*Corresponding author

Abstract—Recent research has shown that deep learning techniques outperform traditional steganography and steganalysis methods. As a result, researchers have proposed increasingly complex and more extensive convolutional Neural Networks (CNNs) to detect Steganographic images to achieve a 1%–2% improvement over the state-of-the-art. In this paper, we propose a data preprocessing and distribution strategy that enhances accuracy and convergence during training. Our method involves bifurcating Spatial Rich Model (SRM) and Discrete Cosine Transform (DCT) filters, with one branch being trainable and the other untrainable. This strategy is followed by three blocks of residual convolutions and an excitation layer. Our proposed method improves the accuracy of CNNs applied to steganalysis by 2%–15% while maintaining stability.

Keywords—convolutional neural network, deep learning, steganalysis, steganography, steganographic filters

I. INTRODUCTION

Recently, security challenges in the era of artificial intelligence have become a significant concern in digital transformation and communication [1]. As data and digital information are being transmitted rapidly over public networks, the technology for protecting and securing sensitive messages must be continuously discovered and developed. Digital multimedia steganography is an essential branch of information hiding, which provides a practical and secure way to address the problem of multimedia communication security. Steganography aims to hide messages in digital media such as images, audio, video, and text [2].

Steganography is formulated with the famous problem of the prisoners (Alice and Bob), who want to exchange messages under the watchful eye of the prison warden. If

the prison warden considers the message suspicious, she will not allow it to reach the recipient [3]. Image steganography is applied in the frequency and spatial domains [4]. In the spatial domain, the Least Significant Bit (LSB) of each pixel in the image is changed, adapting to the outline of the image so that it is invisible to the human eye [5, 6]. The most common transform in the frequency domain is the Discrete Cosine Transform (DCT), with Joint Photographic Experts Group (JPG) data compression. DCT coefficients are modified to include the hidden message. Steganography algorithms in the spatial domain are HILL [7], HUGO [8], S-UNIWARD [9], WOW [10], and MiPOD [11]. For the frequency domain, there are algorithms such as F5 [12], J-UNIWARD [9], UERD30 [13], and UED [14].

Image steganalysis consists of detecting the hidden messages within an image [15, 16]. Initially, detecting hidden messages within an image was performed with machine learning techniques. Machine learning generated problems due to the separation of the two processing stages (feature extraction and steganographic image classification). The feature extraction stage was usually performed with the Rich model [17], and the classification was performed with a support vector machine [18] and an ensemble classifier. If information is lost in the feature extraction stage, it is not recovered in the classification stage because they are two separate processes. Deep Learning (DL) techniques [19] and GPUs [20] improved the classification rate of steganographic images. DL techniques unify feature extraction and image classification stages into a single model eliminating errors generated with a support vector machine and ensemble classifier. In particular, Convolutional Neural Networks (CNNs) is a powerful tool for classifying steganographic images in frequency and spatial domains.

Before CNNs in steganalysis, the most advanced approach was Spatial Rich Models (SRM) [17]. SRM is a

Manuscript received February 14, 2023; revised March 28, 2023; accepted May 19, 2023; published January 9, 2024.

typical steganalysis method in which a high-pass filter bank is designed to obtain the steganographic image's noise residuals. Image residuals are quantized and truncated by reducing their dynamic range. The SRM feature consists of high-order co-occurrence matrices collected from the noise residuals of the steganographic image. In SRM, symmetry rules were designed to improve feature robustness and reduce feature dimensions. SRM shows two significant characteristics; the difference between cover-stego image characteristics is minimal, and different cover images' characteristics are often very other. The first characteristic means the distance between classes is small, while a large dispersion in the same class characterizes the second. Another widely used technique is to generate DCTR features as a result of performing convolution of the image in the spatial domain with the 64 DCT base patterns [21]. It has improved the results when classifying JPEG stego and cover images. In this study, we concatenate the two methods of SRM and DCTR features followed by a residual block stage and an activation layer.

Main steganalysis architectures in chronological order for the spatial domain are the Xu-Net [22], Ye-Net [23], Yedroudj-Net [24], SR-Net [25], Zhu-Net [26], and GBRAS-Net [27]. The above CNNs, except for Xu-Net and SR-Net, use SRM filter banks to assist the feature extraction stage.

This paper presents an exhaustive experimentation process in which different forms of image preprocessing on steganalysis CNNs were tested to determine which is most relevant, then design an image preprocessing stage that improves the accuracy of steganographic image detection for multiple architectures. SRM and DCT bifurcation was developed to improve the accuracy of the CNNs by 1% up to 10% approximately. This paper aims to generate a preprocessing module capable of transforming old CNNs into state-of-the-art competitive CNNs, without excessively increasing the computational cost. Moreover, this preprocessing will be the starting point for future research.

The rest of the paper is structured as follows: Section II describes the most relevant CNNs in image steganalysis in the spatial domain. Section III gives the details of materials, databases, Spatial Rich Models, experiments, Training, hyperparameters, Hardware, and resources. Section IV describes the results and Section V discusses the results. Finally, Section VI presents the conclusions of the work.

II. RELATED LITERATURE

This section will describe in chronological order the most relevant CNNs in steganalysis in the spatial domain:

A. Xu-Net

We use the Xu-Net with strategy in [28] for this research. Xu-Net is a CNN developed by Xu *et al.* [22]. This CNN consists of 5 convolutional layers in the feature extraction stage. The input is convolved with the 30 SRM filters in the preprocessing block. Subsequently, the activation layer $3 \times \text{TanH}$ is used. This neural network

with strategy incorporates a Spatial Dropout from the second convolutional block, implements a Leaky Rectified Linear Unit (ReLU) activation layer, after activation uses an Absolute (ABS) layer, has Batch Normalization (BN) after ABS, and a layer that concatenates three inputs. The classification stage has three Fully Connected (FC) layers of 128, 64, and 32 units, respectively, with a Leaky ReLU activation and a SoftMax layer. The classification block is after global average pooling, and the optimizer was stochastic gradient descent.

B. Ye-Net

This CNN was proposed by Ye *et al.* [23]. For this research, Ye-Net with strategy in [28] was used. Like the previous networks, it implements a convolution to the input data with the 30 SRM filters to increase the perception of stenographic noise. It also has an activation function of $3 \times \text{TanH}$. This CNN has eight convolutional layers for the feature extraction stage. With the strategy implemented in Ref. [28], this network has Spatial Dropout from the second convolutional block, activation Leaky ReLU, ABS, and BN in the above order. It has a concatenation layer of 3 inputs. In the classification stage, 3 FC with 128, 64, and 32 units were activated with Leaky ReLU and SoftMax. The classification stage is after global average pooling and optimized with stochastic gradient descent.

C. Yedroudj-Net

This efficient type of CNN was designed by Yedroudj *et al.* [24], a hybrid of the best features of the convolutional neural networks Xu-Net, and Ye-Net. For this research, Yedroudj-Net was implemented with improvements to the strategy developed by Tabares-Soto *et al.* [28]. The architecture of this convolutional neural network implements the 30 SRM filters, implementing a $3 \times \text{TanH}$ activation. This architecture has five convolutional layers in the feature extraction stage, and from the second layer, it implements Spatial Dropout, uses Leaky ReLU as an activation function, uses ABS, BN, and a 3-input concatenation layer. The classification stage uses three fully connected layers of 128, 64, and 32 units, respectively, with Leaky ReLU and SoftMax activation. The classification stage is after global average pooling and uses stochastic gradient descent as an optimizer.

D. GBRAS-Net

Developed by Reinel *et al.* [27], this architecture has a $3 \times \text{TanH}$ activation function and convolves the input image with 30 SRM filters in a preprocessing block. The activation function chosen for the convolution of the feature extraction stage is Exponential Linear Unit (ELU). This architecture also has separable convolutions, depth separable convolutions, and shortcuts for feature extraction, which handle the same padding and the same number of filters at the beginning and the end. The trigger function chosen for the convolution of the feature extraction stage is ELU. This architecture also has separable convolutions, depth separable convolutions,

and shortcuts for feature extraction, which handle the same padding and the same number of filters at the beginning and the end. This architecture implements separable convolutions inside the shortcuts with some filters of 30 and 60 of size 3×3 , with a stride of 1×1 and no change in padding. This CNN consists of 8 convolutional layers with 3×3 filters, ending this stage has two convolutional layers of size 1×1 , a stride of 1, and does not change the padding. The first two layers of this stage are convolved with 30 filters, the following four layers with 60 filters, the penultimate layer with 30 filters, and the last one with 2. This CNN implements average pooling layers after batch normalization to reduce dimensionality, with a pooling size of 2×2 and a stride of 2. This network uses a SoftMax layer after global average pooling and does not use fully connected layers.

III. MATERIALS AND METHODS

A. Database

Experiments were performed with the BOSSBase 1.01 database, which has 10,000 grayscale images of size $512 \times 512 \times 1$ pixels in a Portable Gray Map (PGM) [29]. The images were resized to $256 \times 256 \times 1$ pixels. The stego images were created from each cover image using the steganographic algorithms S-UNIWARD [9] and WOW [10] with two different payloads (0.4 and 0.2 bpp). The open-source tool Aletheia [30] was used to apply steganography to the images. The images were split into cover and stego pairs, of which 4,000 was used for training, 1,000 for validation, and 5,000 for testing. Data distribution was based on [22, 23, 26].

B. Spatial Rich Models (SRM)

The Spatial Rich Models (SRM) [17] are a set of sub-models that capture various types of relationships among neighboring samples of noise residuals from steganographic images. Constructing these sub-models starts by computing the noise residuals using high-pass filters, which suppress image content allowing for a more robust statistical description. Then, truncation and quantization are applied to these noise residuals, truncation limits the residual dynamic range, and quantization makes the residual more sensitive to changes in edges and textures. After that, co-occurrence matrices are computed in vertical and horizontal directions that are later turned into sub-models by leveraging symmetries of natural images [17].

Since their introduction in 2012, SRM has been essential in steganalysis systems. Even for the latest CNN contributions, this set of filters acts as a preprocessing stage that enhances the steganographic noise improving the detection accuracy. Fig. 1 presents the set of SRM used for digital image steganalysis.

C. DCTR Feature Extraction

DCTR is a process developed by Holub *et al.* [21], which they expose in more depth in their research, below we show roughly how these feature maps are obtained. The DCTR features are obtained by convolving the image

in the spatial domain with the 64 DCT base kernels. Each has a size of 8×8 , giving us an activation map of 64 filters with dimensions 256×256 .

$$B^{(k,l)} = (B_{mn}^{(k,l)}), 0 \leq m, n, k, l \leq 4: \\ B_{mn}^{(k,l)} = \frac{w_k w_l}{4} \cos \frac{\pi k(2m+1)}{16} \cos \frac{\pi l(2n+1)}{16} \quad (1)$$

where $w_0 = \frac{1}{\sqrt{2}}$, $w_i = 1$ for $i > 0$.

The grayscale image X is convolved with each of the 64 DCT basis patterns $B(k, l)$, to generate a set of 64 undecimated DCTs, each of which is denoted by $U(k, l)$ for the (k, l) -th DCT basis pattern as set of 64 DCTs, each of which is denoted by $U(k, l)$ for the (k, l) -th DCT basis pattern as:

$$U^{(k,l)} = X \cdot B^{(k,l)}, 0 \leq k, l \leq 4 \quad (2)$$

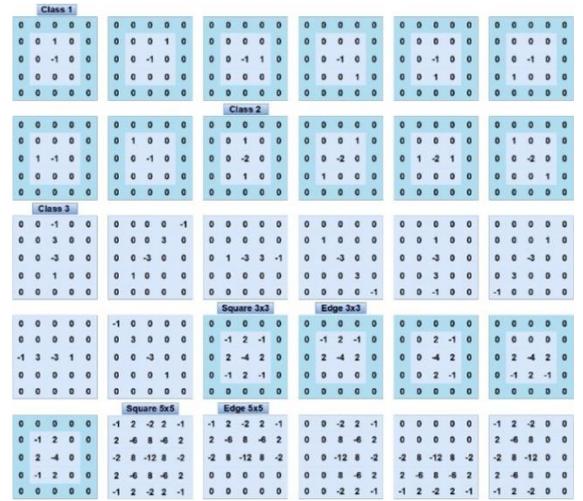


Fig. 1. The 30 SRM filter values.

D. Strategy Proposed to Improving the Preprocessing Stage

The preprocessing stage is crucial in the feature extraction process because it is the stage where the filters highlight the steganographic noise. In this research, a bifurcation was designed in the preprocessing stage. Each branch concatenates the 30 SRM filters mentioned in Section III-B and 64 DCT basis weights mentioned in Section III-C, giving a total of 94 filters for each branch of the bifurcation, one of these branches will be kept trainable (the weights of the 94 filters are modified in training) and the other branch is kept constant, see Fig. 2. The 94 output channels of each branch are summed to form a 94-channel feature map. In SR-Net [25], it is discussed that clustering can reduce the power of steganographic noise, but the shortcut connection can benefit the signal from this noise. Using the same strategy as in [25, 31], a so-called bottleneck block and two basic blocks based on residual networks without pooling [32]. The bottleneck performs a 1×1 convolution with the 94 output filters of the bifurcation, followed by another 3×3 convolution and a final 1×1 convolution. An attention module called ‘‘Squeezeand-Excitation’’ [33] was added to perform a direct access connection with the features

obtained by the bifurcation. Two basic residual building blocks are implemented. Each stage consists of 3×3 convolutions and 94 kernels, see Fig. 3. In all residual blocks, no pooling was performed, and a Batch Normalization (BN) layer and Leaky ReLU activation were used.

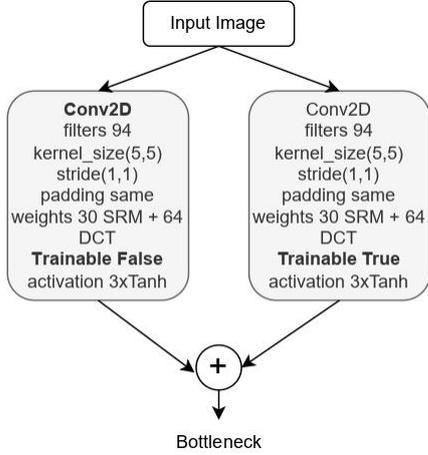


Fig. 2. A preprocessing stage, bifurcation of concatenated, trainable, and untrainable SRM and DCT filters.

E. Training and Hyper-Parameters

1) *GBRAS-Net*: For this network, a batch size of 32 images was set and trained for 150 epochs. All layers were initialized with normal glorot except the initial

preprocessing layer. The Adam optimizer was used with the following configuration:

- Learning rate: 0.001
- Beta 1: 0.9
- Beta 2: 0.999.
- Decay: 0.0
- Epsilon: 1×10^{-8}

This network uses a categorical cross-entropy loss for two classes, a batch normalization with momentum 0.2, epsilon 0.001, scale is False, center is True, trainable True, fused is None, renorm clipping is None, renorm is False, adjustment is None and renorm momentum is 0.4. the maximum absolute value normalizes the SRM filters.

2) *Others CNNs*: The training batch size for the Xu-Net, Ye-Net, and Yedroudj-Net networks was 32 images and trained for 150 epochs. The hyperparameters used for the fully connected layers and the convolutional layers are as follows:

- Normal glorot initializer.
- L2 regularization for kernels and bias.
- The spatial dropout rate has a value of 0.1.
- BN has a momentum value of 0.2.
- Epsilon: 0.001.
- Renorm: 0.4.
- The momentum of the stochastic gradient descent optimizer: 0.95.
- Learning rate initialized: 0.005.

The activation used for the convolutional layers is ReLU with a negative slope of 0.1, converting the ReLU to a leaky ReLU.

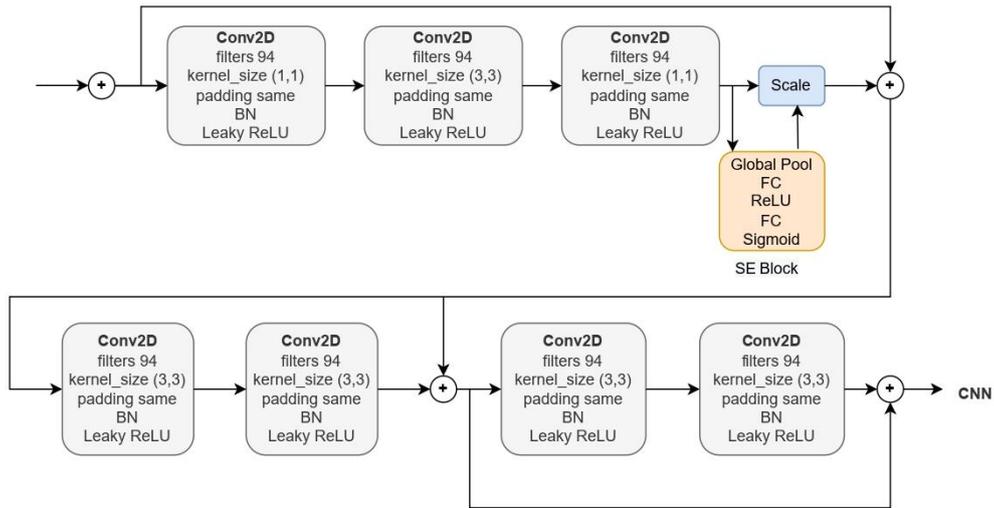


Fig. 3. Proposed residual neural network with 94 channels as input (SRM + DCT Bifurcation), the output is the input of the neural networks trained in this research.

F. Hardware and Resources

Python 3.7.5 was used to build the architecture for this experiment, and the model was designed mainly with TensorFlow 2.4.1 [34] on a workstation running Ubuntu as an operating system. The computer runs 2 Tesla V100-PCIE 32 GB graphics cards, CUDA Version 11.2, 7 Intel(R) Xeon(R) Gold 6130 processors at 2.10 GHz, and 40GB of RAM. For design and initial experimentation,

the Google Colaboratory platform was used in an environment with a Tesla P100 PCIe with 16 GB, CUDA Version 10.1, and 25 GB of RAM.

All resources, including source code and databases of this project, are available as open-source software in the following repository: <https://github.com/BioAITeam/Pre-processing-Strategy-to-Improve-the-Performance-of-Convolutional-Neural-Networks-Applied-to-Stegan>.

IV. RESULT

Figs. 4 and 5 show the learning curves as a function of training epochs for the GBRAS-Net, Ye-Net, and Xu-Net networks for the S-UNIWARD and WOW steganographic algorithms, respectively, trained for 150 epochs.

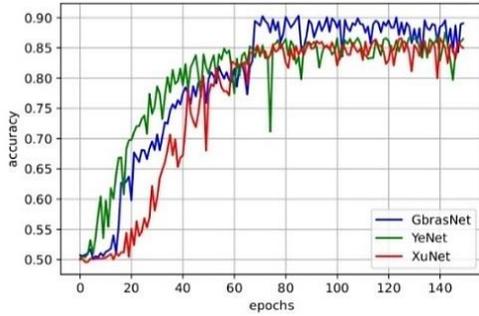


Fig. 4. Test accuracy Curves for S-UNIWARD steganographic algorithms with a 0.4 bpp payload of GBRAS-Net, Ye-Net and Xu-Net with our strategy.

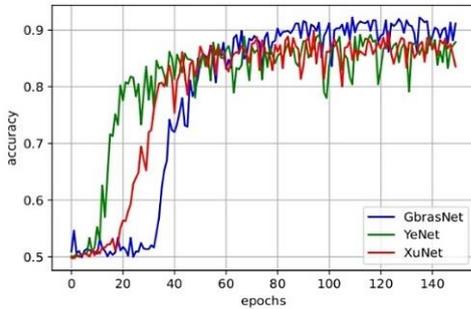


Fig. 5. Test accuracy Curves for WOW steganographic algorithms with a 0.4 bpp payload of GBRAS-Net, Ye-Net and Xu-Net with our strategy.

This preprocessing strategy was trained on the BOSSBase 1.01 database to improve the accuracy of the CNNs spoken in Section II, using S-UNIWARD and WOW as steganographic algorithms at a payload of 0.2 bpp and 0.4 bpp.

The results obtained for this research on the CNN Xu-Net are compared with the results achieved by this network in its original form and with the improvement made by Tabares-Soto *et al.* [28], see Table I.

TABLE I. PERCENT ACCURACY VALUE FOR XU-NET COMPARED TO STRATEGIES FOR THE S-UNIWARD AND WOW STEGANOGRAPHIC ALGORITHMS, WITH PAYLOADS OF 0.2 AND 0.4 BPP

CNN	S-UNIWARD 0.2 bpp	S-UNIWARD 0.4 bpp	WOW 0.2 bpp	WOW 0.4 bpp
Original Xu-Net	60.90	72.80	67.60	79.30
Xu-Net with strategy in [28]	68.29	78.19	73.52	82.21
Xu-Net with our strategy	75.96	87.54	79.91	89.89

Table II shows the results of this research compared to the original CNN, the modification made by Tabares-Soto *et al.* [28] and our strategy.

TABLE II. PERCENTAGE ACCURACY VALUE IN YE-NET COMPARED TO STRATEGIES FOR THE S-UNIWARD AND WOW STEGANOGRAPHIC ALGORITHMS, WITH PAYLOADS OF 0.2 AND 0.4 BPP

CNN	S-UNIWARD 0.2 bpp	S-UNIWARD 0.4 bpp	WOW 0.2 bpp	WOW 0.4 bpp
Original Ye-Net	60	68.80	66.90	76.80
Ye-Net with strategy in [28]	71.03	81.01	75.47	84.51
Ye-Net with our strategy	75.96	87.54	80.71	89.17

The results of Yedroudj-Net for the experiment are compared with the original network and with the improvement made by Tabares-Soto *et al.* [28], see Table III.

TABLE III. PERCENTAGE ACCURACY VALUE IN YEDROUDJ-NET COMPARED TO STRATEGIES FOR THE S-UNIWARD AND WOW STEGANOGRAPHIC ALGORITHMS, WITH PAYLOADS OF 0.2 AND 0.4 BPP

CNN	S-UNIWARD 0.2 bpp	S-UNIWARD 0.4 bpp	WOW 0.2 bpp	WOW 0.4 bpp
Original Yedroudj-Net	63.30	77.20	72.20	85.90
Yedroudj-Net with strategy in [28]	67.73	79.64	76.23	84.70
Yedroudj-Net with our strategy	76.23	86.79	79.56	89.26

The last experiment was performed on GBRAS-Net, Table IV compares the results of this experiment compared to the original network.

TABLE IV. PERCENT ACCURACY VALUE IN GBRAS-NET COMPARED TO STRATEGY FOR THE S-UNIWARD AND WOW STEGANOGRAPHIC ALGORITHMS, WITH PAYLOADS OF 0.2 AND 0.4 BPP

CNN	S-UNIWARD 0.2 bpp	S-UNIWARD 0.4 bpp	WOW 0.2 bpp	WOW 0.4 bpp
Original GBRAS-Net	73.60	87.10	80.30	89.80
GBRAS-Net with our strategy	79.91	91.00	85.90	92.23

TABLE V. PERCENT ACCURACY VALUE IN GBRAS-NET WITH OUR STRATEGY COMPARED TO THE CONVOLUTIONAL TRANSFORMER (CVT) FOR THE WOW STEGANOGRAPHIC ALGORITHM, WITH PAYLOADS OF 0.2 AND 0.4 BPP

CNN	WOW 0.2 bpp	WOW 0.4 bpp
CVT stego	85.25	92.10
GBRAS-Net with our strategy	85.90	92.23

In recent years significant advances have been developed in artificial intelligence thanks to implementing transformers. Research in steganalysis has not been the exception. For this research, we take the results achieved by Luo *et al.* [35], who developed an architecture of convolutional transformers improving many of the results reported in the literature on traditional convolutional architectures. Table V compares GBRAS-Net with our strategy and the results achieved by Luo *et al.* [35]. Table VI shows the percentage increase in

accuracy of the steganalysis architectures applying our strategy compared to the original ones.

TABLE VI. PERCENTAGE INCREASE IN ACCURACY OVER THE ORIGINAL NETWORKS USING OUR STRATEGY ON THE PROPOSED CNNs FOR THE S-UNIWARD AND WOW STEGANOGRAPHIC ALGORITHMS WITH A PAYLOAD OF 0.2 BPP AND 0.4 BPP

CNN	S-UNIWARD	S-UNIWARD	WOW	WOW
	0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp
Xu-Net	15.06	14.74	12.31	10.59
Ye-Net	15.96	18.74	13.81	12.37
Yedroudj-Net	12.96	9.59	7.36	3.36
GBRAS-Net	6.31	3.90	5.60	2.43

Fig. 6 shows the ROC curve for GBRAS-Net, with an AUC of 0.98 for both classes, indicating excellent model performance on the binary classification task.

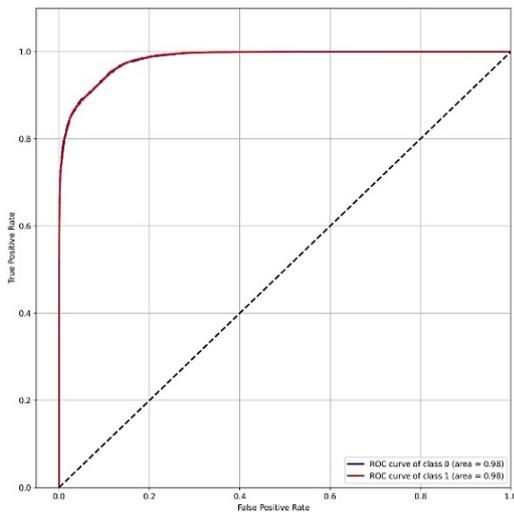


Fig. 6. ROC curve for GBRAS-Net with our strategy in WOW 0.4 bpp.

V. CROSS VALIDATION

Machine learning models sometimes fail to generalize adequately or produce significantly varying results when changing the test data. Therefore, dividing data into training, validation, and test sets does not always have an objective result, as it may incur bias or overfitting. In the experiment presented in Table VII, 10-fold cross-validation is employed to evaluate the proposed strategy, and the reported accuracy and standard deviation provide relevant information on the model’s generalization capability.

TABLE VII. METRICS CALCULATED IN THE 10-FOLD CROSS-VALIDATION FOR CNNs APPLYING THE STRATEGY FOR THE STEGANOGRAPHIC ALGORITHMS S-UNIWARD AND WOW WITH A PAYLOAD OF 0.4 BPP

CNN	S-UNIWARD 0.4 bpp		WOW 0.4 bpp	
	Acc	SD	Acc	SD
Xu-Net	87.14	2.81	89.38	2.15
Ye-Net	87.26	2.61	88.94	1.84
Yedroudj-Net	86.13	1.95	89.35	1.88
GBRAS-Net	90.65	1.83	91.83	1.61

VI. DISCUSSION

Since the introduction of CNN for image steganalysis, the main contribution of the field has been the design of new architectures that further improve detection accuracy. This paper presents a modification of the preprocessing stage of some existing CNNs and three residual blocks with an attention channel, improving accuracy without significant changes in the provided architectures. Based on the results shown in Tables I–IV, the bifurcation and the inclusion of a trainable set of SRM filters, DCT filters, and residual blocks improve the accuracy of steganographic image detection. The main idea of this fork is to leverage the original SRM and DCT filters and allow the network optimization process to update these filters based on the content of the images, which increases the perception of steganographic noise. In addition, the main idea of including the three residual blocks and the attention channel is to save energy, avoid gradient fading, and focus on the steganographic noise. The results indicate that concatenating SRM and DCT filters, making them trainable adds variability and increases the efficiency of the preprocessing stage, perhaps by focusing on different image features. Table VI shows the percentage increase in accuracy provided by the strategy concerning the original networks.

As shown in Figs. 4 and 5, the strategy not only increases the accuracy percentage when classifying stego and cover images but also preserves the stability of the networks, as the test accuracy during training evolves similarly in all cases.

VII. CONCLUSION

The proposed preprocessing stage has improved the detection of steganographic images without significant changes to the original CNN architectures. The bifurcation of SRM and DCT filters into trainable and untrainable, along with including three combined residual blocks called bottleneck with an attention block (SE-Block) and two residual building blocks, has demonstrated the critical role of data preprocessing in steganalysis. The modified preprocessing stage has improved the perception of steganographic noise and facilitated training and convergence in feature extraction, thereby achieving higher accuracy.

The results suggest that optimizing DCT and SRM filters can further increase the efficiency of data preprocessing, improving steganalysis accuracy. The high complexity of CNNs may not be necessary if efficient data preprocessing is available. The proposed strategy increases the model’s sensitivity to steganographic noise thanks to the preprocessing stage, allowing it to focus globally on different image characteristics. Figs. 4 and 5 demonstrate that this strategy improves accuracy in classifying stego and cover images and preserves network stability. Future research could focus on optimizing DCT and SRM filters to enhance the efficiency of data preprocessing and steganalysis accuracy.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

MABO, HBAA, GI, RRP, and RTS conceived the experiments; EMR, JPVP, and MATM conducted the experiments; MABO and EMR analyzed the results; MABO, EMR, JPVP, HBAA, GI, RRP, MATM, and RTS wrote and reviewed the manuscript. All authors had approved the final version.

FUNDING

This research was funded by Dirección de Investigaciones y Posgrados of Universidad de Manizales.

ACKNOWLEDGMENT

The authors thank the Universidad Autonoma de Manizales, Manizales, Colombia, for support and covering publication fees under project 645-2019 TD.

Also, the authors would like to thank the Dirección de Investigaciones y Posgrados of the Universidad de Manizales for their support.

REFERENCES

- [1] M. Hassaballah, M. A. Hameed, A. I. Awad, and K. Muhammad, "A novel image steganography method for industrial internet of things security," *IEEE Trans. Industr. Inform.*, vol. 17, no. 11, pp. 7743–7751, 2021.
- [2] M. A. Hameed, O. A. Abdel-Aleem, and M. Hassaballah, "A secure data hiding approach based on least-significant-bit and nature-inspired optimization techniques," *J. Ambient Intell. Humaniz. Comput.*, 2022.
- [3] G. J. Simmons, "The prisoners' problem and the subliminal channel," *Advances in Cryptology*, Boston, MA: Springer US, 1984, pp. 51–67.
- [4] M. Hassaballah, *Digital Media Steganography: Principles, Algorithms, and Advances*, San Diego, CA: Academic Press, 2020.
- [5] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *Computer (Long Beach Calif.)*, vol. 31, no. 2, pp. 26–34, 1998.
- [6] J. Fridrich, M. Goljan, and R. Du, "Detecting LSB steganography in color, and gray-scale images," *IEEE Multimed.*, vol. 8, no. 4, pp. 22–28, 2001.
- [7] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," presented at 2014 IEEE International Conference on Image Processing (ICIP), 2014.
- [8] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," *Information Hiding*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 161–177.
- [9] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP J. Multimed. Inf. Secur.*, vol. 2014, no. 1, 2014.
- [10] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," presented at 2012 IEEE International Workshop on Information Forensics and Security (WIFS), 2012.
- [11] V. Sedighi, R. Cogranne, and J. Fridrich, "Content-adaptive steganography by minimizing statistical detectability," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 2, pp. 221–234, 2016.
- [12] A. Westfeld, "F5—A steganographic algorithm: High capacity despite better steganalysis," in *Proc. Information Hiding*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 289–302.
- [13] L. Guo, J. Ni, W. Su, C. Tang, and Y.-Q. Shi, "Using statistical image model for JPEG steganography: Uniform embedding revisited," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 12, pp. 2669–2680, 2015.
- [14] L. Guo, J. Ni, and Y. Q. Shi, "Uniform embedding for efficient JPEG steganography," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 5, pp. 814–825, 2014.
- [15] T.-S. Reinel, R.-P. Raul, and I. Gustavo, "Deep learning applied to steganalysis of digital images: A systematic review," *IEEE Access*, vol. 7, pp. 68970–68990, 2019.
- [16] R. Tabares-Soto *et al.*, "Digital media steganalysis," *Digital Media Steganography*, Elsevier, 2020, pp. 259–293.
- [17] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 3, pp. 868–882, 2012.
- [18] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [19] S. Theodoridis, "Neural networks and deep learning," *Machine Learning*, Elsevier, 2020, pp. 901–1038.
- [20] R. T. Soto, *Parallel Programming on Heterogeneous Architectures*, 2016. (in Spanish)
- [21] V. Holub and J. Fridrich, "Low-complexity features for JPEG steganalysis using undecimated DCT," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 2, pp. 219–228, 2015.
- [22] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 708–712, 2016.
- [23] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 11, pp. 2545–2557, 2017.
- [24] M. Yedroudj, F. Comby, and M. Chaumont, "Yedroudj-Net: An efficient CNN for spatial steganalysis," presented at 2018 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2018.
- [25] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 5, pp. 1181–1193, 2019.
- [26] R. Zhang, F. Zhu, J. Liu, and G. Liu, "Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 1138–1150, 2020.
- [27] T.-S. Reinel *et al.*, "GBRAS-Net: A convolutional neural network architecture for spatial image steganalysis," *IEEE Access*, vol. 9, pp. 14340–14350, 2021.
- [28] R. Tabares-Soto *et al.*, "Strategy to improve the accuracy of convolutional neural network architectures applied to digital image steganalysis in the spatial domain," *PeerJ Comput. Sci.*, vol. 7, e451, 2021.
- [29] P. Bas, T. Filler, and T. Pevný, "Break our steganographic system: The ins and outs of organizing BOSS," *Information Hiding*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 59–70.
- [30] D. L. Hostalot, "Daniel Lerch's personal page," *Daniellerch.me*, 2023.
- [31] W. You, H. Zhang, and X. Zhao, "A Siamese CNN for image steganalysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 291–306, 2021.
- [32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," presented at 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- [33] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," arXiv preprint, arXiv:1709.01507, 2017.
- [34] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," arXiv preprint, arXiv:1603.04467, 2016.
- [35] G. Luo, P. Wei, S. Zhu, X. Zhang, Z. Qian, and S. Li, "Image steganalysis with convolutional vision transformer," presented at ICASSP 2022, 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022.

Copyright © 2024 by the authors. This is an open access article distributed under the Creative Commons Attribution License ([CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.