Received: August 6, 2023. Revised: August 29, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Pixel-block-based Steganalysis Method for Hidden Data Location in Digital Images

Ntivuguruzwa Jean De La Croix[1,2] **Tohari Ahmad[1]*** **Royyana Muslim Ijtihadie[1]**

[1]*Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia*
[2]*African Center of Excellence in Internet of Things, University of Rwanda, Kigali, 3900, Rwanda*
* Corresponding author's Email: tohari@if.its.ac.id

Abstract: Concealing confidential data within digital multimedia, also called steganography, has been crucial for safeguarding data. However, steganalysis techniques are best known for compromising these data concealment methods. The convolutional neural network (CNN) yielded better performance over the previously suggested techniques based on machine learning for steganalysis. Although the current approaches show promising outcomes, they encounter challenges in locating the secret data. This paper introduces a CNN architecture with a key idea to enhance the accuracy of identifying the precise hidden data location. Our method novelty is based on reducing image features and network parameters via customized feature allocation and convolutional kernels optimization and enhancing classification accuracy by optimizing the kernels combined with back-forward gradient descent. Our experimental results show improvements over the existing works with the accuracy under payload capacity of 0.4 bpp from 83.01% to 98.78% for S-UNIWARD and 87.19% to 97.93% for WOW.

Keywords: Information security, Steganography, Steganalysis, Convolutional neural network, Network infrastructure.

1. Introduction

Image steganography is a technique used in secret communication to hide confidential data within ordinary cover media such as images [1], video [2], audio [3], and text [4]. In digital images, the modified pixel values in stego images are subtly altered, making them difficult to discern from the original images [5–8]. With image steganalysis, the objective is to identify the traces of embedding or even attempt to recover the concealed information in a reverse steganography process [9], as illustrated in Fig. 1, which presents the general logic connecting steganography and steganalysis paradigms. Image steganalysis can be summarized in three may classes as follows. Determining a given image's class (cover or stego) [10]; subsequently, the identified suspicious images, known as stego images, are selected for further investigation. In the case of the known steganographic method, the analysis can be expanded to include estimating the payload size, which is referred to as quantitative steganalysis [11]. Moreover, the third class of steganalysis, known as locative steganalysis, entails revealing the locations of the hidden secret message [12].

Since the embedding key is typically unavailable, a commonly employed alternative is to implement a method that identifies the most probable locations of the hidden payloads, such as in blocks of the pixels [13] or regions [14] of a suspicious image. Given the absence of orderings such as logical ones, the steganalysis schemes are limited to detecting the payloads rather than explicitly revealing the locations of the secret messages. Nevertheless, locating the small groups of the image pixels with the hidden data holds vital significance.

Most researchers in the field of steganalysis have primarily recently tried to concentrate on identifying the existence of hidden data in the content of digital images [12, 15, 16]. Departing from the introduction of the rich models, the state-of-the-art works showed outperforming results in detecting the presence of hidden data in the content of the digital images.

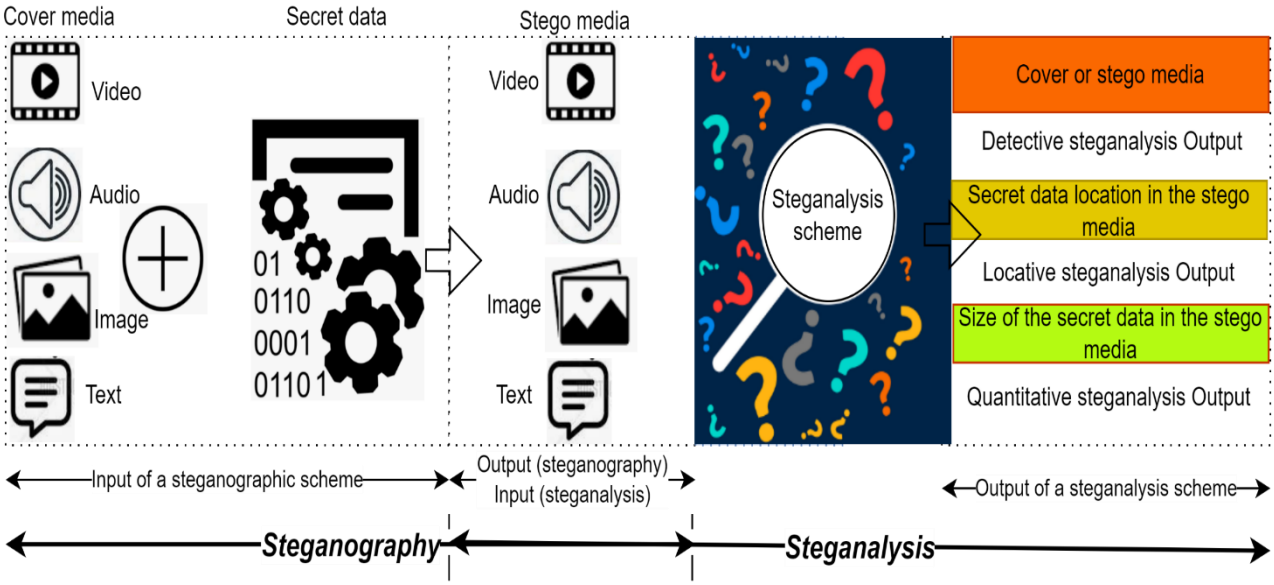

Figure. 1 Steganography and steganalysis processes illustration

Table 1. Notation list

Symbol	Description
cov	Cover image
d	Euclidean distance
$f_{conceal}$	Function to conceal the secret data
FN	False negative
FP	False positive
L_m	Length of the generated bitstream
m	Random bitstream
S	Stego image
t_m	Total pixels in the cover image
TN	True negative
TP	True positive
β	Relative payload
μ	Neighboring weight

Researchers in detective steganalysis have recently tried to improve the results of steganalysis schemes in locating the confidential bits confiscated in the digital images by using various deep learning approaches combined with other mathematical models such as fuzzy logic [10], neighboring weight algorithm [16], or by using pixels orders manipulation for the most effective regions with the secret data detection [14]. Moreover, some other studies tried to improve confidential data location accuracy by combining convolutional neural networks (CNNs) for locating the secret bits hidden through emerging steganography algorithms [14].

The existing solutions in addressing the locative steganalysis problem showed a significant contribution with high accuracy; nevertheless, they present several drawbacks such as high computing capacity (see [14]), accuracy that still need to be improved to optimize the schemes correctness for the fields that may be very sensitive to any error such military applications, medical, and forensic applications [17]. To alleviate the problems identified from the existing solutions, this research proposes an enhanced CNN, a deep learning (DL) scheme with optimized parameters, by reducing the convolutional kernels' sizes and splitting the inquiry image into pixel blocks. To enhance the readability of our equations and the work in general, we add Table 1, containing a notation list. The contribution of this study is described as follows:

(1) Reducing the size of the images for our model to optimize the feature allocation, improving the hidden data location accuracy.
(2) Reducing the number of parameters by adapting the convolutional kernels and initializing them with the basic 30 SRM filter banks as of [18].
(3) Optimizing the convolutional kernel to improve the accuracy with optimal convergence of the CNN using the back-forward gradient descent, otherwise known as the back-forward slope.

The remaining parts of this article are organized in the following sections. Section 2 summarizes the state-of-the-art models that have been most popular in locative steganalysis, section 3 includes the description of our method, section 4 encompasses the experimental results with their analysis, and section 5 gives a summative inference of the paper.

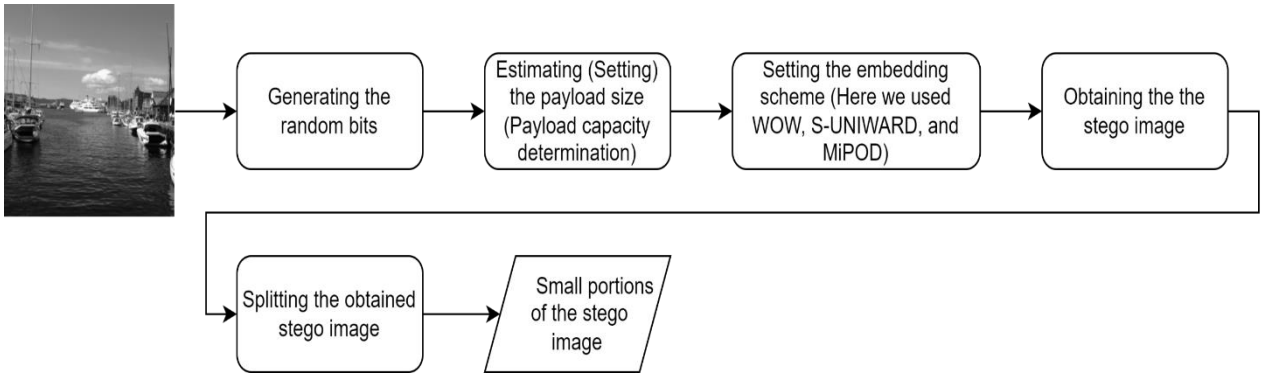

Figure. 2 The process to get the blocks from the stego image

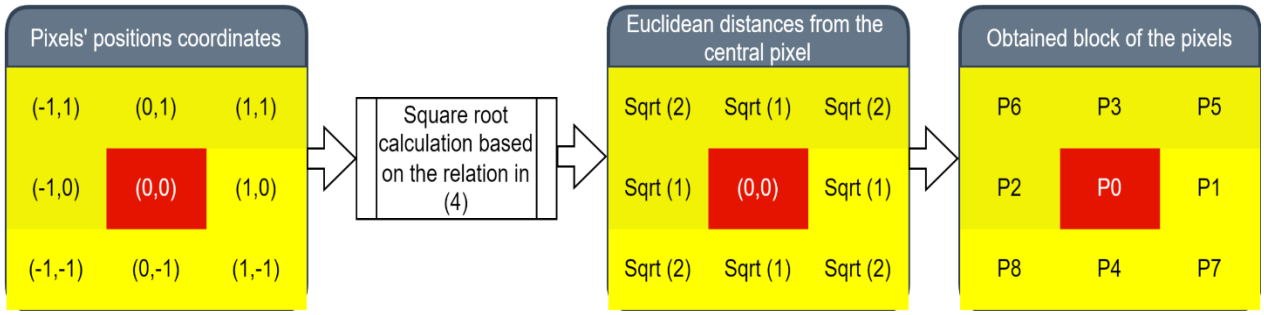

Figure. 3 The blocks formation process

Legend: (a) The red colour shows the data attributed to the central pixel and (b) The yellow colour shows the data about the neighboring pixels constituting the block

2. Existing literature

Highlighting the payload locations continues to be a crucial aspect of research in image steganalysis, which mainly lies in the information security field. Here, we provide a brief overview of existing efforts to locate the presence of the secret within digital images.

In the research conducted in [19], a proposed approach addresses the issue of pixel classification by categorizing each pixel into two binary types: payload and non-payload. This research work achieved promising results in steganographic payload location by exploring the features of each pixel. During locating hidden bits, a support vector machine (SVM) classifier is trained using discriminative features. These features consist of 72-dimensional characteristics that capture the differences in pixel values between neighboring pixels. The SVM classifier is then employed for binary classification purposes. Nevertheless, this technique's drawback is based on the fact that though the accuracy and efficiency of the locating process are enhanced by the learning-based method, the performance experiences degradation as the steganographic payload is increased.

In [20], a detector believed to be efficient has been proposed to locate the secret data hidden in digital images based on a deep neural network. However, this approach is currently limited to its applicability only to stego images generated using older steganography techniques, such as nonadaptive LSBM. Moreover, in [21], a study demonstrated a paved way for exploring the detection of steganographic payloads concealed using modern adaptive algorithms. Their method was mainly based on re-embedding the secret bits in the pre-confirmed stego images to predict the locations double concealed with the secret data to be the actual locations of the confidential data. The drawback of this method is based on a degraded recall and precision rate to locate the steganographic payload, which is identified through the yielded F1-score, which is still lower than 0.5. This shows that this method still needs to be improved regarding payload location accuracy.

In addition, several other research works proposed methods believed to be more efficient in locating the pixels modified with the addition of confidential data, such as in [12, 16]. The work in [16] suggested a payload location method based on the neighboring weight algorithm (NWA). The proposed method introduces a novel algorithm to

identify the steganographic payload in the spatial domain by beginning by making predictions about the data embedding algorithm and its associated payload, which are then utilized to generate random bitstreams of data. Subsequently, the previously generated random bits are concealed within the firstly confirmed stego image using a cost matrix within the syndrome-trellis codes (STCs) scheme to get a second stego image. By leveraging the discrepancies between two stego images, the authors could obtain the extended modification map using the neighboring weight algorithm. Their process ultimately facilitated the identification and localization of the hidden bits. The drawbacks of this method are identified in pairs, namely, the loss of discriminative features because the proposed process might discard certain relevant details, reducing the capability to distinguish steganographic images effectively, and the significant sensitivity of the size of the payload because the payload size incrementation may affect the locative ability because of the altered visual features.

Moreover, in [12], a method to locate the secret data by combining fuzzy logic and CNN. The proposed approach for locating the secret data in a digital image has been structured into three stages. Firstly, they start by computing the modification maps between the stego and the cover images. Secondly, these modification maps are utilized as input for the fuzzy inference system, which employs four input membership functions (distance vector, covariance map, compass mean, and the intensity of pixels matrices) and the membership function for output, namely the correlation maps from the fuzzy inference system. Finally, the fuzzy-resulted maps known as correlation maps are fed into a CNN to distinguish pixels altered by secret data concealment from the original pixels. Through experimental analysis conducted on four adaptive steganographic models (HILL, HUGO-BD, S-UNIWARD, and WOW), the results substantiate their superior performance. The drawback of this method is based on the fact that their approach did not address the problem of broad generalization across diverse steganographic algorithms or scenarios, potentially resulting in compromised accuracy.

In line with the existing works, differently to [12, 16], that preprocess the inquiry images, which may result in misclassification due to losing some significant features, we consider non-pre-processed inquiry images for input to our model. Moreover, to address the remarkable problem of the results deterioration problem when the payload is increased as of [19, 21], our method considers splitting an inquiry image into small parts, which are easy and effective for classification. Furthermore, we mitigate the issue of less generalization discussed in [20] by making our model work with adaptive steganographic algorithms. We propose a new steganalysis algorithm to locate the hidden data without requiring much training because we work on the blocks (where one image generates several blocks) and a Simple CNN with small kernel sizes.

3. Proposed method

In this section, we present the steps involved in the algorithm we present in locating a steganographic payload by initially embedding the secret data in the cover images, splitting the inquiry images into blocks, and classifying the blocks based on whether they hold the secret data or not. Fig. 2 illustrates the steps taken to get the blocks from the stego images, and Algorithm 1 details the overall process to locate the blocks with the hidden data.

From Algorithm 1, here is the description of the steps:

1) *Creating a random sequence of bits*
Denoting the random bitstream as m, we generate m with a specified length L_m.
Considering t_m as the cover image's total pixels and β as the relative payload, we find the measure of L_m basing on Eq. (1).

$$L_m = t_m \times \beta \tag{1}$$

2) *Concealing the secret data (here referred to as random bits)*
We use the generally known scheme known as syndrome trellis codes (STCs) [22] to embed the secret data. Considering cov as the cover, we obtain the stego images here denoted as S as of Eq. (2).

$$S = cov + m \tag{2}$$

3) *Splitting the stego and cover into blocks*
Considering a pixel P of coordinates (i_o, j_o) As a central pixel, we depart from the neighboring weight factor μ, got from the Euclidean distance d between the pixel P_1 of coordinates (i_1, j_1) near the central pixel P to form the pixel blocks for the training process. The neighboring weight is as of Eq. (3), and the Euclidean distance is calculated following Eq. (4).

$$\mu = \sqrt{d} \tag{3}$$

Table 2. A summarized description of the proposed CNN architecture

Type of the layer	**Output shape**	**Parameters**	**Linked to**
input_1 (Input Layer)	[(None, 256, 256, 1)]	0	[ ]
2D Convolution (2D Conv)	(None, 256, 256, 30)	780	['input_1[0][0]']
Batch Normalization (BN)	(None, 256, 256, 30)	90	['2d_conv[0][0]']
DepthwiseConv2D	(None, 256, 256, 30)	60	['bn[0][0]']
SeparableConv2D	(None, 256, 256, 30)	3540	['depthwise_2d_conv [0][0]']
Batch Normalization (BN)	(None, 256, 256, 30)	90	['separable_2d_conv [0][0]']
DepthwiseConv2D	(None, 256, 256, 30)	60	['bn_1[0][0]']
SeparableConv2D	(None, 256, 256, 30)	3540	['depthwise_2d_conv _1[0][0]']
Batch Normalization (BN)	(None, 256, 256, 30)	90	['separable_2d_conv _1[0][0]']
add (Add)	(None, 256, 256, 30)	0	[' bn[0][0]', 'bn_2[0][0]']
2D Convolution (2D Conv)	(None, 256, 256, 30)	8130	['add[0][0]']
Batch Normalization (BN)	(None, 256, 256, 30)	90	['2d_conv _1[0][0]']
2D Convolution (2D Conv)	(None, 256, 256, 30)	8130	['bn_3[0][0]']
Batch Normalization (BN)	(None, 256, 256, 30)	90	['2d_conv_2[0][0]']
Average Pooling2D	(None, 128, 128, 30)	0	['bn_4[0][0]']
2D Convolution (2D Conv)	(None, 128, 128, 60)	16260	['average_pooling2d[0][0]']
Batch Normalization (BN)	(None, 128, 128, 60)	180	['2d_conv_3[0][0]']
DepthwiseConv2D	(None, 128, 128, 60)	120	['bn_5[0][0]']
SeparableConv2D	(None, 128, 128, 60)	12480	['depthwise_2d_conv _2[0][0]']
Batch Normalization (BN)	(None, 128, 128, 60)	180	['separable_2d_conv_2[0][0]']
DepthwiseConv2D	(None, 128, 128, 60)	120	['bn_6[0][0]']
SeparableConv2D	(None, 128, 128, 60)	12480	['depthwise_2d_conv_3[0][0]']
Batch Normalization (BN)	(None, 128, 128, 60)	180	['separable_2d_conv_3[0][0]']
add_1 (Add)	(None, 128, 128, 60)	0	['bn_5[0][0]"bn_7[0][0]']
2D Convolution (2D Conv)	(None, 128, 128, 60)	32460	['add_1[0][0]']
Batch Normalization (BN)	(None, 128, 128, 60)	180	['2d_conv _4[0][0]']
AveragePooling2D	(None, 64, 64, 60)	0	['bn_8[0][0]']
2D Convolution (2D Conv)	(None, 64, 64, 60)	32460	['average_pooling2d_1[0][0]']]
Batch Normalization (BN)	(None, 64, 64, 60)	180	['2d_conv _5[0][0]']
Average Pooling2D	(None, 16, 16, 60)	0	['bn_10[0][0]']]

(Continued) Table 2. A summarized description of the proposed CNN architecture

Type of the layer	Output shape	Parameters	Linked to
2D Convolution (2D Conv)	(None, 16, 16, 30)	1830	['average_pooling2d_3[0][0]']
Batch Normalization (BN)	(None, 16, 16, 30)	90	['2d_conv_7[0][0]']
2D Convolution (2D Conv)	(None, 16, 16, 2)	62	['bn _11[0][0]']
Batch Normalization (BN)	(None, 16, 16, 2)	6	['2d_conv _8[0][0]']
Global Average Pooling2D	(None, 2)	0	['bn_12[0][0]']
Softmax	(None, 2)	0	['global_average_pooling2d[0][0]'
Total parameters: 166,598 Trainable parameters: 164,734 Non-trainable parameters: 1,864			

Algorithm 1

Input: Cover image cov, random bitstream m

Output: Prediction of the blocks with the steganographic payload.

Step 1: Creating a random sequence of bits L_m

$$L_m = t_m \times \beta$$

with tm: the cover image's pixels

β: relative payload capacity

Step 2: Concealing the secret data to get the stego image S.

$$S = f_{conceal}(cov, L_m)$$

with $f_{conceal}$, a function used to conceal the secret data L_m into cov in the STCs framework.

Step 3: Splitting the stego images S and the cover images cov into blocks based on the logic in Fig. 3 which relies on the Euclidean distance.

Step 4: Padding the block with ones ($1s$) to have dimension 256×256.

Step 5: Classification of the padded blocks into innocent or altered (holding the steganographic payload) blocks using the CNN detailed in Table 2.

$$d = \sqrt{(i_1 - i_0)^2 + (i_1 - i_0)^2} \quad (4)$$

Considering one pixel at each side of the neighborhood of the central pixel, we will form the blocks of size (3,3) in this work. The block formation is also illustrated in Fig. 3.

4) Padding the block with ones (1s) to have dimensions 256×256

Our blocks initially dimensioned to 3×3, and we use NumPy to create a new array with dimension 256×256 and then put the original block into the centre of the new array, surrounded by ones.

5) Binary classification of the blocks into innocent or altered pixel blocks using a CNN

Using a CNN, we optimize the feature allocation and the number of model parameters through the convolutional kernels' initialization with the SRM filters. To improve the model's convergence, we use the back-forward descent gradient. The general structure of the CNN we propose is summarized in Table 2, including the layer types, input and output shapes, parameters, and the connection fashion of the layers.

4. Results

4.1 Experimentation setup

This study carefully defines the experimental settings to comprehensively evaluate and compare the steganographic schemes and the locating method.

The proposed approach involves experiments using the BOSSbase ver.1.01 dataset [23]. This dataset comprises 10,000 grey images obtained from eight distinct cameras. Each image has a size of 512×512 pixels and was stored in an uncompressed format to ensure the preservation of data integrity. The payload for the steganographic algorithms tested in the experiment is $0.2\ bpp$ and $0.4\ bpp$. The steganographic schemes tested in this study include WOW, S-UNIWARD, and MiPOD. Our method was also utilized for comparison alongside existing locating methods [12, 21]. These well-defined experimental settings provided a solid foundation for conducting a rigorous analysis of the proposed scheme and its effectiveness in locating the pixels altered by a steganographic payload.

Furthermore, to conduct a comprehensive evaluation of our method's performance, the accuracy, precision, recall rate, otherwise known as sensitivity and the $F_1-\text{score}$ has been used. The computation of the $F_1-\text{score}$ departs from the Recall rate here computed as of Eq. (5) and the precision computed as of Eq. (6). The accuracy is obtained based on the relation Eq. (7) and the $F_1-\text{score}$ is got based on Eq. (8).

The accuracy represents the proportion of correctly classified blocks over the total number f considered blocks. In the context of our model, the accuracy evaluates how accurately a block with embedded data is located. Precision represents the proportion of the true positive predictions to the total positive predictions of blocks. For the case of this work, the precision shows the accuracy of positively located blocks out of all blocks predicted to be positive. Recall rate is also referred to as sensitivity or true positive rate. The recall rate, or recall, represents the ratio of true positively predicted blocks to the total positive blocks demonstrated as holding the secret data. The F1-score balances precision and recall rate, which plays a capital role if an imbalance between the classes happens. In the context of steganalysis, particularly this work, a higher F1-score identifies a better trade-off between correctly located steganographic data and minimizing false positives cases.

It is worth noting that the abbreviations used in our evaluation metrics are defined as follows: true positives (TP) represent the number of changed pixels correctly identified as altered pixels; false positives (FP) indicate the number of cover pixels incorrectly identified as altered pixels; true negatives (TN), indicate the number of cover's pixels correctly identified as cover pixels; and false negatives (FN), represent the stego pixels incorrectly identified as cover pixels.

$$Recall\ rate = \frac{TP}{FN+TP} \tag{5}$$

$$Precision = \frac{TP}{FP+TP} \tag{6}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \tag{7}$$

$$F_1 - score = 2 \times \frac{Precision \times Recall\ rate}{Precision+Recall\ rate} \tag{8}$$

4.2 Results and discussion

This subsection represents the results of the proposed model evaluation on the training, validating, and testing process. It also presents a comparative analysis of the obtained results and the results in the existing works.

1) Model evaluation of the training process

The data in Table 3 shows the proposed model's training process results over different steganographic algorithms using specific payload capacities (expressed in bits per pixel, bpp) and various evaluation metrics, namely accuracy, precision, recall, and F1-score.

It is worth noting that as the payload capacity increases, the performance to detect all three algorithms is generally improved. The WOW algorithm consistently demonstrates outperforming results in all metrics considered for evaluation and indicates the same across all payload capacities. The MiPOD algorithm also shows significant performance, particularly at 0.4 bpp. In contrast, the S_UNIWARD algorithm generally shows lower results than WOW and MiPOD, particularly regarding the recall rate and F1-score.

2) Model evaluation on the validation process

Referring to Table 4, which contains the results of our method's validation process, it is identified that the performance is consistently improved across all algorithms. Specifically, it is worth noting that our algorithm performs best to locate the blocks holding data embedded using the WOW compared to other steganographic algorithms. Locating the steganographic data hidden with the MiPOD algorithm achieves commendable results, mainly when the payload capacity is higher at 0.4 bpp, where it maintains competitive performance across the considered evaluation metrics. Meanwhile, the location of the payload concealed under S_UNIWARD, while generally delivering respectable outcomes, shows slightly inferior results in all evaluation

Table 3. Results of the training process

Steganographic algorithm	Payload capacity (bpp)	Evaluation metrics			
		Accuracy (%)	Precision (%)	Recall (%)	F1-score
MiPOD	0.2	96.64	96.43	95.75	0.96
	0.4	98.52	98.41	97.85	0.98
S_UNIWARD	0.2	96.02	95.81	95.13	0.95
	0.4	98.01	97.90	97.34	0.97
WOW	0.2	98.55	98.44	97.76	0.98
	0.4	98.73	98.52	98.37	0.98

Table 4. Results of the validation process

Steganographic algorithm	Payload capacity (bpp)	Evaluation metrics			
		Accuracy (%)	Precision (%)	Recall (%)	F1-score
MiPOD	0.2	97.44	97.33	96.68	0.97
	0.4	99.23	98.60	98.49	0.98
S_UNIWARD	0.2	96.82	96.71	96.05	0.96
	0.4	98.72	98.09	97.98	0.98
WOW	0.2	99.92	99.88	99.02	0.99
	0.4	99.97	99.93	99.27	0.99

Table 5. Results of the testing process

Steganographic algorithm	Payload capacity (bpp)	Evaluation metrics			
		Accuracy (%)	Precision (%)	Recall (%)	F1-score
MiPOD	0.2	96.67	95.65	94.99	0.95
	0.4	98.49	98.33	97.66	0.97
S_UNIWARD	0.2	96.04	95.03	94.37	0.94
	0.4	97.93	97.82	97.16	0.97
WOW	0.2	97.89	97.78	97.12	0.97
	0.4	98.78	98.67	98.01	0.98

metrics compared to the other two algorithms. Our evaluation results generally showcase our model's efficiency in locating the image's blocks with steganographic payload with promising accuracy, precision, recall, and F1-score.

3) Model evaluation on the testing process

Table 5 contains the results of our model for the testing process where the model best locates the block holding a steganographic payload concealed under the WOW algorithm with notable stability across both 0.2 and 0.4 bpp, with a significant performance in terms of Accuracy, Precision, Recall, and F1-score. MiPOD also demonstrates commendable performance, with accuracy, precision, recall rate, and F1-score steadily enhancing in direct proportions as the payload capacity. Similar performance improvement in direct proportion to the payload size is observed for S_UNIWARD, which also maintains promising results showing relatively modest output in both the Recall rate and the F1-score.

4.3 Comparison of our results to the state-of-the-art

In Table 6, we present a comparison of the results obtained with our model and the results of the state-of-the-art models in [12, 21], considering the accuracy, precision, recall, and F1-score as key evaluation metrics. To obtain the comparison result from [12] in terms of accuracy, we depart from a table presenting their results with several payload capacities (we multiply the considered data by 100 to express them in the same scale as our work); in terms of the recall rate and F1-score, we depart a systematic observation and mathematical interpretation of figures illustrated in their paper. We convert the obtained values to the same scale as ours to have homogeneous data for comparison. To get the comparison data from [21], we also systematically observe and mathematically interpret the illustrated figures for both the recall rate and the F1-score in their work. To optimize the trade-off between the recall rate and the precision and extract the optimal

Table 6. Results comparison between our method and the existing methods

Method	Steganographic method	Payload capacity (bpp)	Evaluation metrics		
			Accuracy (%)	Recall (%)	F1-score
The proposed Method	WOW	0.2	96.04	94.37	0.94
		0.4	97.93	97.16	0.97
	S_UNIWARD	0.2	97.89	97.12	0.97
		0.4	98.78	98.01	0.98
The method in [12]	WOW	0.2	69.98	91.00	0.22
		0.4	87.19	94.50	0.35
	S_UNIWARD	0.2	62.88	95.00	0.17
		0.4	83.01	96.50	0.23
The method in [21]	WOW	0.2	-	73.00	0.27
		0.4	-	81.00	0.34
	S_UNIWARD	0.2	-	57.00	0.16
		0.4	-	70.00	0.23

data, we consider the recall rate with the feature map margin yielding the best outcomes in F1-score and set the extracted values to the same scales as ours.

It is worth noting that the method we propose in this work demonstrates outperforming results either with a payload of 0.2 bpp or 0.4 bpp achieving high scores in terms of Accuracy, Recall, and F1-score. The methods proposed [12, 21] exhibit varying performance levels, but it is generally shown that our method yields the best results. It is also worth noting that the start-of-the-art works considered did not work on detecting the location of the steganographic payload hidden under the MiPOD algorithm as done in our work. The findings generally underscore the proposed method's superior performance compared to existing approaches, positioning it as a promising advancement in locative steganalysis techniques.

5. Conclusion

Research has been conducted to enhance steganalysis performance in locating the steganographically altered parts of digital images with hidden data. The introduction of the CNNs demonstrated their ability to yield remarkable results surpassing the conventional ML-based handcrafted features. This article centres around the CNN paradigm to develop a new CNN that employs reduced features and optimized kernels to detect the pixel blocks containing hidden data within images.

Compared to the existing schemes, our contributions are as follows. First, our model improves the accuracy of detecting hidden data by optimizing feature allocation by reducing the image size. Second, we reduce the number of parameters by customizing the convolutional kernels and initializing them with the fundamental 30 SRM filter banks, as presented in [18]. Third, we enhance the accuracy and achieve optimal convergence of the CNN by optimizing the convolutional kernel using backpropagation.

For experimentation, we utilize images from the BOSSBase version 1.01 dataset with the recent adaptive steganographic algorithms, namely, WOW, S-UNIWARD, and MiPOD. We embed random data with payload capacities of 0.2 bpp and 0.4 bpp. Our experimental findings exhibit a notable performance of our method compared to existing techniques in accurately locating the pixel blocks containing steganographic payload within digital images.

In future works, our objective is to extend the application of this proposed method to other datasets, including agricultural and medical images, thereby making a cross-field contribution to the detection of hidden data presence which can help the decision-makers or security measures accuracies.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, NJDLC, TA and RMI; methodology, NJDLC, TA and RMI; software, NJDLC; validation, NJDLC, TA and RMI; investigation, NJDLC; resources, TA; writing—original draft preparation, NJDLC; writing—review and editing, TA; visualization, NJDL; supervision, TA and RMI; project administration, TA; funding acquisition, TA.

Acknowledgements

The authors thank all the laboratory and research group members for their valuable contributions and support.

This research received support from the Ministry of Education, Culture, Research and Technology, Republic of Indonesia, and Institut Teknologi Sepuluh Nopember.

References

[1] N. J. D. L. Croix, C. C. Islamy, and T. Ahmad, "Secret Message Protection using Fuzzy Logic and Difference Expansion in Digital Images", In: *Proc. of the 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development, NIGERCON 2022, Institute of Electrical and Electronics Engineers Inc.*, 2022. doi: 10.1109/NIGERCON54645.2022.9803151.

[2] T. Ahmad and A. N. Fatman, "Improving the performance of histogram-based data hiding method in the video environment", *Journal of King Saud University - Computer and Information Sciences*, Vol. 34, No. 4, pp. 1362–1372, 2022, doi: 10.1016/j.jksuci.2020.04.013.

[3] I. B. Prayogi, T. Ahmad, N. J. D. L. Croix, and P. Maniriho, "Hiding Messages in Audio using Modulus Operation and Simple Partition", In: *Proc. of 2021 13th International Conference on Information and Communication Technology and System, ICTS 2021, Institute of Electrical and Electronics Engineers Inc.*, pp. 51–55, 2021. doi: 10.1109/ICTS52701.2021.9609028.

[4] B. Khosravi, B. Khosravi, B. Khosravi, and K. Nazarkardeh, "A new method for pdf steganography in justified texts", *Journal of Information Security and Applications*, Vol. 45, pp. 61–70, 2019, doi: 10.1016/j.jisa.2019.01.003.

[5] N. J. D. L. Croix, C. C. Islamy, and T. Ahmad, "Reversible Data Hiding using Pixel-Value-Ordering and Difference Expansion in Digital Images", In: *Proc. of IEEE International Conference on Communication, Networks and Satellite, COMNETSAT 2022, Institute of Electrical and Electronics Engineers Inc.*, pp. 33–38, 2022, doi: 10.1109/COMNETSAT56033.2022.9994516.

[6] I. Théophile, N. J. D. L. Croix, and T. Ahmad, "Fuzzy Logic-based Steganographic Scheme for high Payload Capacity with high Imperceptibility", In: *Proc. of 2023 11th International Symposium on Digital Forensics and Security (ISDFS), IEEE*, pp. 1–6, 2023, doi: 10.1109/ISDFS58141.2023.10131727.

[7] A. J. Ilham, T. Ahmad, N. J. D. L. Croix, P. Maniriho, and M. Ntahobari, "Data Hiding Scheme Based on Quad General Difference Expansion Cluster", *Int J Adv Sci Eng Inf Technol*, Vol. 12, No. 6, p. 2288, 2022, doi: 10.18517/ijaseit.12.6.16002.

[8] M. S. Hossein, T. Ahmad, and N. J. D. L. Croix, "Data Hiding Scheme using Difference Expansion and Modulus Function", In: *Proc. of 2023 2nd International Conference for Innovation in Technology (INOCON)*, pp. 1–6, 2023 doi: 10.1109/INOCON57975.2023.10100991.

[9] G. Xie, J. Ren, S. Marshall, H. Zhao, R. Li, and R. Chen, "Self-attention enhanced deep residual network for spatial image steganalysis", *Digit Signal Process*, Vol. 139, p. 104063, 2023, doi: 10.1016/j.dsp.2023.104063.

[10] N. J. D. L. Croix and T. Ahmad, "Toward Hidden Data Detection via Local Features Optimization in Spatial Domain Images", In: *Proc. of 2023 Conference on Information Communications Technology and Society (ICTAS), IEEE*, pp. 1–6, 2023, doi: 10.1109/ICTAS56421.2023.10082736.

[11] C. Rupa, S. Shaikh, and M. Chinta, "Multimedia concealed data detection using quantitative steganalysis", *International Journal of Digital Crime and Forensics*, Vol. 13, No. 5, pp. 101–113, 2021, doi: 10.4018/IJDCF.20210901.oa6.

[12] N. J. D. L. Croix and T. Ahmad, "Toward secret data location via fuzzy logic and convolutional neural network", *Egyptian Informatics Journal*, Vol. 24, No. 3, p. 100385, 2023, doi: 10.1016/j.eij.2023.05.010.

[13] X. Han and T. Zhang, "Spatial Steganalysis Based on Non-Local Block and Multi-Channel Convolutional Networks", *IEEE Access*, Vol. 10, pp. 87241–87253, 2022, doi: 10.1109/ACCESS.2022.3199351.

[14] D. Hu, Q. Shen, S. Zhou, X. Liu, Y. Fan, and L. Wang, "Adaptive Steganalysis Based on Selection Region and Combined Convolutional Neural Networks", *Security and Communication Networks*, Vol. 2017, 2017, doi: 10.1155/2017/2314860.

[15] B. Pan, T. Qiao, J. Li, Y. Chen, and C. Yang, "Novel Hidden Bit Location Method towards JPEG Steganography", *Security and Communication Networks*, Vol. 2022, 2022, doi: 10.1155/2022/8230263.

[16] T. Qiao, X. Luo, B. Pan, Y. Chen, and X. Wu, "Toward Steganographic Payload Location via Neighboring Weight Algorithm", *Security and*

Communication Networks, Vol. 2022, 2022, doi: 10.1155/2022/1400708.

[17] S. Gupta, N. Mohan, and P. Kaushal, "Passive image forensics using universal techniques: a review", *Artif Intell Rev*, Vol. 55, No. 3, pp. 1629–1679, Mar. 2022, doi: 10.1007/s10462-021-10046-8.

[18] J. Fridrich and J. Kodovsky, "Rich Models for Steganalysis of Digital Images", *IEEE Transactions on Information Forensics and Security*, Vol. 7, No. 3, pp. 868–882, 2012, doi: 10.1109/TIFS.2012.2190402.

[19] T. T. Quach, "Optimal cover estimation methods and steganographic payload location", *IEEE Transactions on Information Forensics and Security*, Vol. 6, No. 4, pp. 1214–1222, 2011, doi: 10.1109/TIFS.2011.2160855.

[20] Y. Sun, H. Zhang, T. Zhang, and R. Wang, "Deep neural networks for efficient steganographic payload location", *In Journal of Real-Time Image Processing, Springer Verlag*, pp. 635–647, 2019, doi: 10.1007/s11554-019-00849-y.

[21] Q. Liu, T. Qiao, M. Xu, and N. Zheng, "Fuzzy Localization of Steganographic Flipped Bits via Modification Map", *IEEE Access*, Vol. 7, pp. 74157–74167, 2019, doi: 10.1109/ACCESS.2019.2920304.

[22] T. Filler, J. Judas, and J. Fridrich, "Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes", *IEEE Transactions on Information Forensics and Security*, Vol. 6, No. 3, pp. 920–935, 2011, doi: 10.1109/TIFS.2011.2134094.

[23] P. Bas, T. Filler, and T. Pevný, "Break Our Steganographic System", *The ins and Outs of Organizing BOSS*, pp. 59–70, 2011, doi: 10.1007/978-3-642-24178-9_5.