Ntivuguruzwa and Ahmad *Cybersecurity (2023)6:23*
https://doi.org/10.1186/s42400-023-00156-x

Cybersecurity

RESEARCH

Open Access

A convolutional neural network to detect possible hidden data in spatial domain images

Jean De La Croix Ntivuguruzwa[1,2] and Tohari Ahmad[1*]

Abstract

Hiding secret data in digital multimedia has been essential to protect the data. Nevertheless, attackers with a steganalysis technique may break them. Existing steganalysis methods have good results with conventional Machine Learning (ML) techniques; however, the introduction of Convolutional Neural Network (CNN), a deep learning paradigm, achieved better performance over the previously proposed ML-based techniques. Though the existing CNN-based approaches yield good results, they present performance issues in classification accuracy and stability in the network training phase. This research proposes a new method with a CNN architecture to improve the hidden data detection accuracy and the training phase stability in spatial domain images. The proposed method comprises three phases: pre-processing, feature extraction, and classification. Firstly, in the pre-processing phase, we use spatial rich model filters to enhance the noise within images altered by data hiding; secondly, in the feature extraction phase, we use two-dimensional depthwise separable convolutions to improve the signal-to-noise and regular convolutions to model local features; and finally, in the classification, we use multi-scale average pooling for local features aggregation and representability enhancement regardless of the input size variation, followed by three fully connected layers to form the final feature maps that we transform into class probabilities using the softmax function. The results identify an improvement in the accuracy of the considered recent scheme ranging between 4.6 and 10.2% with reduced training time up to 30.81%.

Keywords Information security, Spatial domain steganalysis, Deep learning, Convolutional neural network, Infrastructure

Introduction

In this cyber era, data transmission within various digital media through public networks plays a capital role in covert communication. With a public network, the communicating sides must ensure security to keep the data private and confidential (Ahmad and Fatman 2022). It is because when the data are accessed unwantedly by a third party, severe security problems may happen, hence research to conceal the secret data in various media types, such as images, audio, and videos, has been carried out (de La Croix et al. 2022b; Prayogi et al. 2021). Concealing data in digital media, also known as steganography, proved an outstanding contribution to covert data transmission in public networks (de La Croix et al. 2022a; Nissar and Mir 2010). The success of steganography served as a valuable enabler to malicious data transmission that can harm society through illegal plan accomplishment. To address the problem of possible covert transmission of harmful data, a counter-steganography technique named steganalysis has been proposed to inspect the integrity of digital media and prevent illegal parties from misusing steganography (Ferreira et al. 2020; Hussain et al. 2020; Tabares-Soto et al. 2021). Figure 1 illustrates the connection between steganography and steganalysis concepts using images.

*Correspondence:
Tohari Ahmad
tohari@if.its.ac.id
[1] Present Address: Department of Informatics, Institut Teknologi Sepuluh Nopember (ITS), Kampus ITS Keputih Sukolilo, Surabaya 60111, Indonesia
[2] African Center of Excellence in the Internet of Things, College of Science and Technology, University of Rwanda, 3900 Kigali, Rwanda

Springer Open

© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

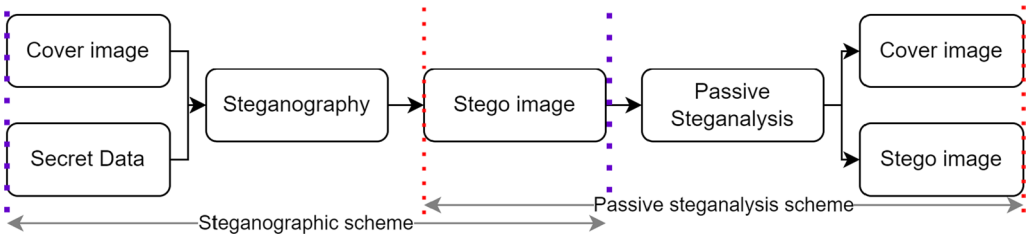

Fig. 1 Steganography and passive steganalysis

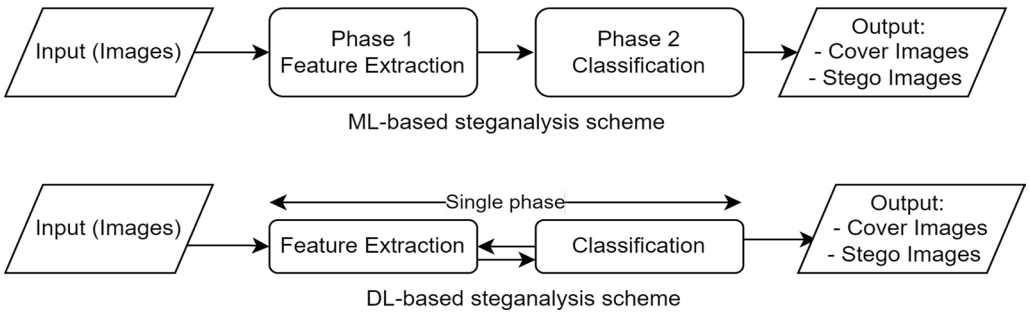

Fig. 2 ML-based versus DL-based steganalysis scheme in digital images

Various ML techniques have been proposed for steganalysis have been proposed (Liu et al. 2020; Lopez-Hernandez et al. 2008; Płachta et al. 2022; Selvaraj et al. 2021; Shankara and Upadhyay 2019) to use two stages namely the feature extraction and classification without backward communication between the stages. The feature extraction phase models the distortion in an image, and the classification phase tries to reveal the existence of a steganographic payload in a particular image and probabilistically concludes whether it is a cover or a stego image (Alsabhany et al. 2020). However, the steganalysis techniques proposed based on ML algorithms did not achieve promising performance for the overall steganalysis tasks due to the general logic of the ML techniques, as illustrated in Fig. 2, which demonstrates the distinction between the ML and deep learning (DL) schemes. Recently, to improve the steganalysis results in general, research works used DL models to design new schemes to perform steganalysis tasks in digital images. DL-based steganalysis schemes include deep neural networks (DNN) and convolutional neural networks (CNN), which unify the feature extraction and classification operations in a single phase (Hussain et al. 2020). The CNN-based methods demonstrated that the feature extraction phase improved the performance of models' generation for image classification (Ikhlayel et al. 2019), which made a meaningful contribution to improving the steganalysis results (Shehab and Alhaddad 2022). However, the previously proposed methods still have gaps in the classification accuracy and training phase stability to be addressed in further research (Rahman et al. 2020; Xiang et al. 2020).

In this paper, we propose a new CNN architecture enlightened by the existing research to improve the performance of the previously proposed steganalysis schemes. Figure 3 illustrates the main phases of the method we propose. We use a pre-processing layer to apply the filters to slice through the pixels of an input image and then apply the traditional and depthwise convolutions with average pooling in the feature extraction phases. In the classification phase, we use multi-scale average pooling under the spatial pyramid pooling (SPP) paradigm, with a succession of three fully connected layers and a softargmax, a normalized exponential function known as the softmax function.

Our CNN achieves better performance in steganographic payload detection accuracy and training stability, and the following characteristics distinguish the proposed CNN from the previously proposed CNNs:

(1) We use two-dimensional (2D) depthwise separable convolutions to prevent issues of kernels that can be skipped for residuals spatial and channel correlation and signal-to-noise ratio (SNR) in the training stage with the main benefit of accuracy enhancement. Moreover, we use this convolution type to prevent overfitting because they have fewer parameters than the traditional convolutional layers.

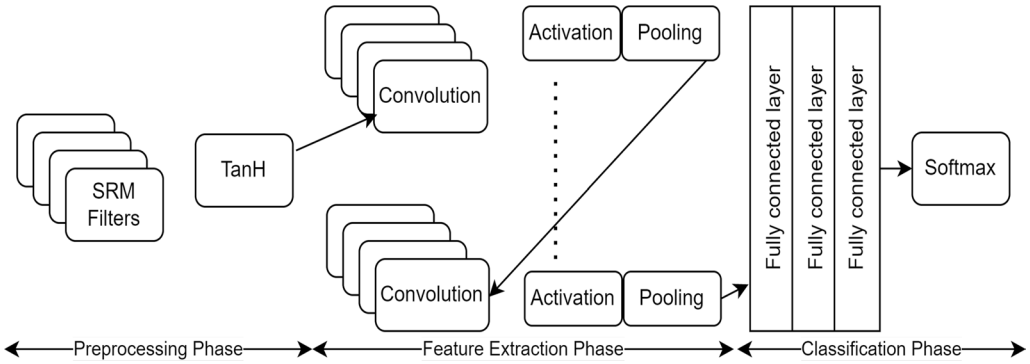

Fig. 3 General architecture of the network with the proposed method

(2) We use the leaky rectified linear unit (LReLu) for non-linearity. The LReLu avoids the vanishing gradients by setting the negative values to small positive values that make backward communication possible by always keeping the model weights positive. Based on the fast convergence achieved with LReLu, the network converges fast and improves the training stability.
(3) We use multi-scale pooling to compact the feature maps regardless of the previous size. The advantage of using multi-scale pooling is to keep the spatial information of various sizes from the earlier layers and enhance feature expression, which contributes to the classification phase.

The following parts of this paper are arranged in sections as follows. "Related works" section discusses the general frameworks of image steganalysis and the related works in spatial domain image steganalysis. We describe our method in "Proposed method" section, and "Experimental setup and results" section presents the experimental setup, obtained results, and their discussion, followed by "Conclusion" section, which concludes our article.

Related works

In this paper, we primarily focus on studying steganalysis in digital images. Generally, the problem of detecting the possible hidden massage in digital images was previously addressed using ML; however, the introduction of DL significantly improved the steganalysis results in digital images. The ML-based steganalysis approaches are known as two-phased approaches because of separating the feature extraction and classification into two phases. Based on Fig. 2 in "Introduction" section, it is identified that the ML-based classification stage consists of using a classifier such as SVM or ensemble classifiers to learn from the feature vector from the feature extraction stage. Though the ML-based steganalysis algorithms achieve promising results, these approaches have improved the detection accuracy by introducing the DL-based approaches.

The introduction of DL-based approaches, namely the DNNs and CNNs to steganalysis (Płachta et al. 2022), tried to improve the performance of the ML-based algorithms by learning the relevant features (Tabares-Soto et al. 2020). Randomly training CNN does not achieve good network convergence and stability performance. Therefore, a steganalysis CNN should be customized to include the domain knowledge for the steganalyzer (Guttikonda & Sridevi 2019). The main design of a steganalysis CNN comprises three modules: the preprocessing module, the feature extraction module, and the classification module (Tereshchenko et al. 2021).

The existing steganalysis CNNs have some features in common in terms of architecture. However, the methods to detect possible secret messages hidden in spatial images differ based on the architectures and the setting of the hyperparameters. This part describes the architectures of the four most recent CNNs for steganalysis, namely the Ye-Net (Ye et al. 2017), Yedroudj-Net (Yedroudj et al. 2018), Zhu-Net (Zhang et al. 2020), and GBRAS-Net (Reinel et al. 2021).

The Ye-Net (2017), enlightened by the previously proposed works, used Spatial Rich Models (SRM) filter banks to pre-process images, eight layers for a convolutional module, and applied Truncation Linear Unit (TLU) as an activation function. To identify the steganographic noise, this model does not implement the traditionally used HPF but implements the HPFs to calculate the SRM residual maps. The results from SRM help in the initialization of the filters. Channel selection knowledge is involved in improving the performance, and an approach known as transfer learning got from training resulted in networks being used to train another network.

Referring to the work of Ye et al., the network by Yedroudj et al. (2018) was implemented by reusing some features of the existing CNNS, such as SRM-based filter banks, TLU, and Batch Normalization (BN) activation functions, and convolutional layers. This model mainly implements the features of Ye-Net and Xu-net by implementing HPF for SRM-based pre-processing. TLU is applied in two layers, and BN is used after all convolutional layers of the next stage after pre-processing, feature extraction. Average pooling is also involved with some convolutional layers.

Inspired by Ye-Net and Yedroudj-Net, a network by Zhang et al. (2020) was presented with the main feature of the reduced filter size for a convolutional layer that decreases the parameters and characteristics of the previous CNNs' architectures. The residual channel and spatial data are correlated using two separable convolutions, image content is compressed, and the SNR is increased. The network availed by Zhu used SRM filter banks for the pre-processing stage. It implemented Spatial Pyramid Pooling (SPP) for feature rendering improvement, adding local features and admission of varied sizes images.

Moreover, with enlightenment from the previously proposed works, (Reinel et al. 2021) presented a CNN with a pre-processing stage of 30 non-trainable SRM filter banks. Using nine layers, CNN applied ELU as an activating function, proving it to be the best-performing network compared to the previous one. In addition, GBRAS-Net implemented two skip connections and four convolutional layers of average pooling of dimension two with stride two. This network introduced eight convolutions, four separable layers, and four depth-wise layers, contributing to the steganographic noise detection efficiency. To prevent the overfitting behaviour with GBRAS-Net, use a global average pooling layer that precedes the use of Softmax. Before the binary classification stage, this network applied a layer for global average pooling.

In this work, we propose a CNN that aims to improve the existing CNNs in detecting the presence of hidden messages in spatial domain images. Our improvement focuses on the convolutional operations that result from the types of layers, non-linearity, and normalization functions to enhance the feature extraction. Moreover, the classification phase focuses on the features modelling and compilation efficiency to improve the classification phase.

Proposed method

Architecture

The proposed CNN focuses on classifying images into one of the two probabilistic classes (stego or cover). The input of our CNN is a digital image of size 256×256, which is manipulated through several layers to be classified as a stego or cover. Referring to an illustration of our CNN in Fig. 4, we start by pre-processing the input image, followed by four depthwise separable convolutional operations, which are performed through two distinct operations per each, namely the depthwise convolutions and pointwise convolutions, and seven regular convolutional layers operations in the feature extraction stage. In the classification stage, we use a multi-scale pooling module to moderate the features to an adaptable size and feed them to three fully connected layers, followed by a softmax function for probability classification.

Pre-processing phase

In the pre-processing phase, we use the small kernels to reduce the modelling regions and the parameters to reduce the calculations. Referring to the design of convolutional kernels in Reinel et al. (2021), which shows that $SQUARE5 \times 5$ and $EDGE5 \times 5$ filters in the SRM filter banks require to be of kernel size 5×5, we adopt two kernel sizes, namely, 5×5, 3×3 in this phase. We keep the kernel size to 5×5 for the $SQUARE5 \times 5$ and $EDGE5 \times 5$ filters and set 3×3 to other 25 SRM high pass filters to model the residual data in the local region. We multiply the residual elements by the filter banks to obtain the pre-processing output channels that serve as input to the feature extraction phase, and the filter slides one step, which is called stride $(1, 1)$. To keep the same size between the input and the output matrices of pixels, we use the padding by zero to fill in the gaps, set as padding the same. To expand the model's possibilities, we apply tangent hyperbolic (3TanH) to serve as a non-linear layer that improves the non-linearity to make our deep network efficient and increase the convergence of the network. The kernel values are not optimized or learned in this stage to target a reduced training time.

Feature extraction phase

In the feature extraction phase, we use the two-dimensional (2D) depthwise separable convolutional combined with regular convolution layers. Taking advantage of depthwise separable convolutions that enhance the model expressiveness by reducing the storage size and separating the correlation between channels, we use four depthwise separable convolutions grouped in two pairs. Based on the general paradigm of the depthwise separable convolutions, (1) is used to express the depthwise convolution set as 3×3 and (2) to express the pointwise convolution set as 1×1. The depthwise convolution captures the channel correlations, and the pointwise convolutional operation captures the spatial

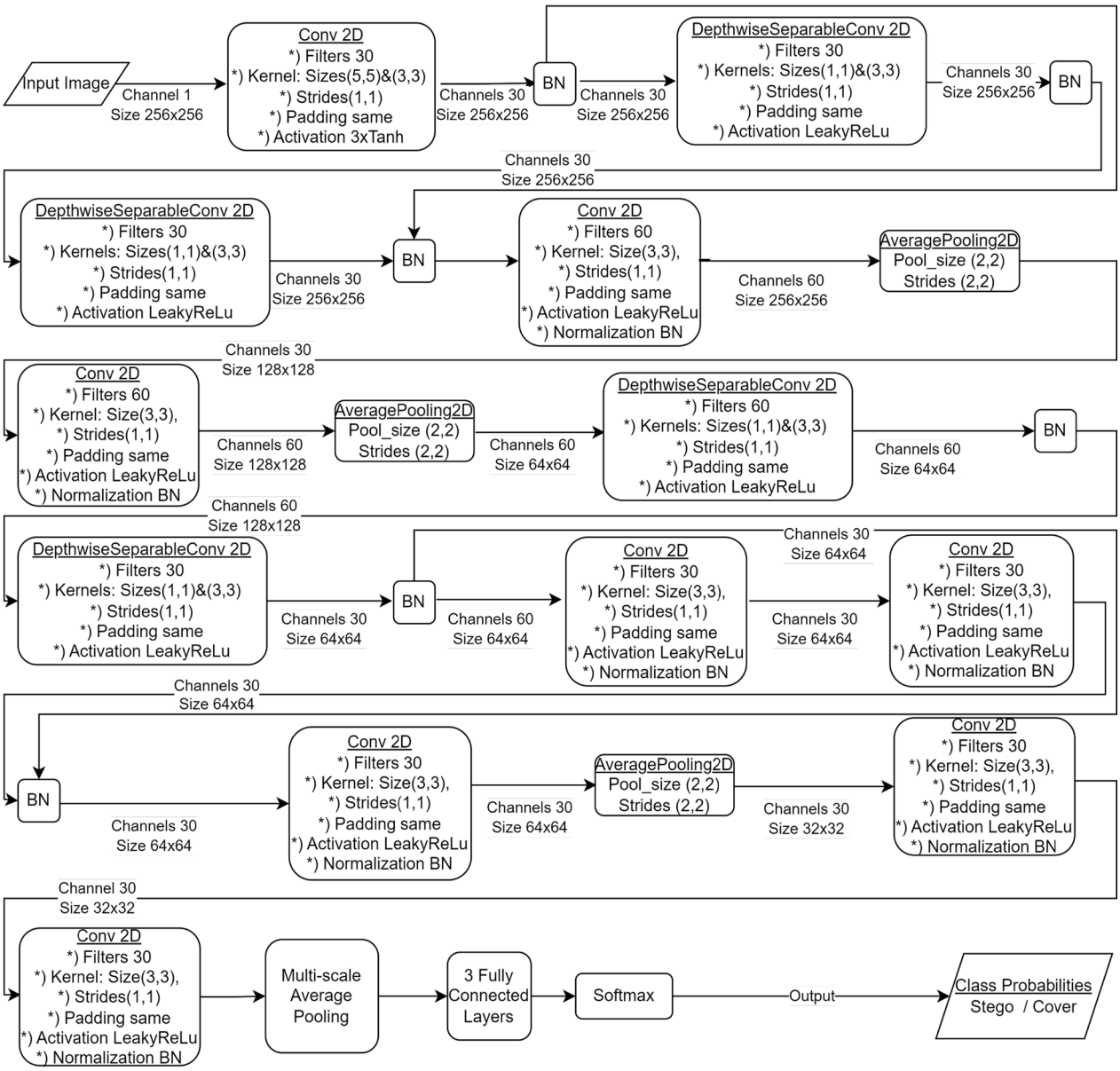

Fig. 4 Schema of the proposed CNN

correlations in the image's residuals. The output feature map F of the depthwise convolution K of size 3×3 is obtained by applying K to each input channel of the feature map I. The final output feature maps of the depthwise separable convolution are equivalent to the output of the pointwise convolution that applies a 1×1 convolution $K^{'}$ to generate a combined output $F^{'}$ of the depthwise convolution.

$$F_{k,l} = \sum K_{i,j} \cdot I_{k+i,l+j} \tag{1}$$

$$F'_{k,l} = \sum K_{m,n} \cdot F_{k-1,l-1} \tag{2}$$

Though the depthwise separable convolutions usually do not need non-linearity for implementation, we improve the learning efficiency by applying the LeakyReLu function as of (3) after the pointwise convolution to improve the non-linearity and a batch normalization (BN) layer to normalize the feature maps distribution. LReLu prevents the vanishing gradients by turning the slopes m with negative values into small but

positive values by multiplying it by a coefficient α, and the BN.

$$m = \begin{cases} mifm \geq 0 \\ \alpha mOtherwise \end{cases}. \quad (3)$$

The regular convolutional layers we use in our CNN apply the kernels of size 3×3 with a slide step of size $(1, 1)$. To keep the same size between the input and output feature maps, we pad the output with zeros by setting the layer as padding the same. To normalize the distribution of the mini batches to unit variance and zero mean in the training phase, we use BN that prevents the vanishing or exploding gradients that may result in overfitting in the network. The BN contributes to increasing the learning rate that speeds up the network convergence.

In addition, as in the Depthwise separable convolutional blocks, we use LReLu in the regular convolutional blocks for non-linearity improvement and to facilitate forward and back propagation. To address the problem of the sensitivity of the output feature maps to locate the features in the input, we use the average pooling operation to reduce the size of the feature maps, known as downsampling. The down-sampled feature maps present more robustness against the changes in the positions of the features, technically known as local translation invariance (Hussain et al. 2020). Moreover, average pooling is preferred because it efficiently enhances the ability of the network to generalize the feature maps.

Classification phase

In the classification phase, we use multi-scale average pooling to model the feature maps to a standard scale from the feature extraction phase, which is crucial for a steganalysis network. To compile the feature maps from the *SPP* module used in modeling through average pooling, we use three successive fully connected layers. The output of the last fully connected layer is supplied to the softmax function to transform the features yielded into the probabilistic classes.

The feature classification phase of our network uses multi-scale pooling, considered an improved approach of SPP to compute the full feature maps of an image and the features in a pool of arbitrary size to generate fixed-length features to be input to the fully connected layers. Like (Zhang et al. 2020), we split the feature maps into different bins, and in each bin, we used the average pooling paradigm to pool the results of feature maps. Our pool is of a type 3-scale pyramid pool of sizes $(4, 4)$, $(2, 2)$, and $(1, 1)$, which makes a total of 21 bins got by $4 \times 4 + 2 \times 2 + 1 \times 1$ for a single feature map. Based on the number n of feature maps generated from the last convolutional layer of the classification phase, we supply to the fully connected layer a vector of size $21 \times n$. We connect the feature maps supplied to the fully connected layers to every activation unit of three successive layers to multiply them by a weight matrix and add a bias vector to form the vector output $\overrightarrow{a_i}$ Which is then converted into probabilistic classes (cover or stego) from the probabilistic distribution of value resulting from the normalized exponential function $\sigma(\vec{a})$ as of (4), otherwise called softmax function.

$$\sigma(\vec{a})_i = \frac{e^{a_i}}{\sum_{j=1}^{k} e^{a_j}} \quad (4)$$

Comparison of our method with the existing methods

To compare our method to the state-of-the-art, in this subsection, we consider the features of the networks proposed in Ye-Net, Zhu-Net, and GBRAS-Net. These three CNNs have been selected because they are among the recent ones and are the most highlighted in the current literature based on their promising performances. The following list shows the similarities and differences between our CNN and the existing CNNs.

(1) The Ye-Net, Zhu-Net, and GBRAS-Net accept input images of the size 256×256. The network we propose also uses input images of the size 256×256, which is the first similarity of our method to the existing ones. For the pre-processing phase, two CNNs, Ye-Net and Zhu-Net, apply 30*SRM* filter banks as trainable filters, and GBRAS-Net uses SRM Filter banks but as non-trainable filters. The proposed CNN adopts the same fashion as GBAS-Net and uses non-trainable 30*SRM* filter banks. The particularity of our method in this phase is that we set the kernel size to 5×5 for the *SQUARE*5×5 and *EDGE*5×5 filters and set 3×3 to other 25 SRM high-pass filters to model the residual data in the local region.
(2) For the feature extraction phase, Ye-Net uses eight convolutional layers, Zhu-Net uses five convolutional layers, GBRAS-Net uses nine convolutional layers, and the proposed CNN uses seven convolutional layers, and depthwise separable convolutions to prevent issues of kernels that can be skipped for residuals spatial and channel correlation, and signal-to-noise ratio (SNR) in the training stage with a primary benefit of accuracy enhancement. Depthwise separable convolutions enhance the model expressiveness by reducing the storage size and separating the correlation between channels, and we use four depthwise separable convolutions.
(3) For the non-linearity function, Ye-Net uses the TLU activation function, Zhu-Net uses ReLU activation,

and GBRAS-Net uses the ELU activation function. The proposed CNN uses LeakyReLU, which outperforms the functions applied in the previous CNNs in two points, the vanishing gradients prevention and permitting forward-backwards communication in the feature extraction phase.

(4) For the classification phase, Ye-Net does not use any pooling operation before its one fully connected layer that supplies its output to the softmax function for probabilistic classification; Zhu-net uses a multi-level average pooling operation and feeds the result to two fully connected layers that send their output to the softmax function for classification into probability classes. GBRAS-net only uses a global average pooling operation and supplies the results to the softmax function for probabilistic classification. In our CNN, we use multi-scale average pooling to model the feature maps to a standard scale from the feature extraction phase, which is crucial for a steganalysis network. To compile the feature maps from the *SPP* module used in modelling through average pooling, we use three successive fully connected layers. The output of the last fully connected layer is supplied to the softmax function to transform the features yielded into the probabilistic classes.

Experimental setup and results

The main components of this section are the experimental setup, experimental results, discussion, results on comparing our model to the state-of-the-art models, results of the cross-dataset experiment, and the ablation study results. The experimental setup subsection includes the datasets selection, hardware, and software resources, hyperparameters setup, and evaluation metrics; the experimental results and discussion subsection discusses the obtained results on the fixed-size state in comparison with the significant existing methods' results; the subsection for the results on the comparison of our model to the state-of-the-art models discusses a comparison between our results and the current methods' results for both the fixed-size images and arbitrary-size images; the results of the cross-dataset experiment subsection discusses the general applicability of our model to verify training phase stability and robustness of our model against overfitting; the ablation study results subsection includes the results achieved by replacing each component of our model to demonstrate the contribution of each element to the performance of our model.

To compare the performance of the proposed method summarized in Fig. 4 with the existing techniques by Reinel et al. (2021), Ye et al. (2017), Yedroudj et al. (2018), Zhang et al. (2020), we adopt their experimental setup for datasets software resources, hyperparameters, and evaluation metrics as reported. Though the problem of steganalysis in images of different sizes is beyond our scope, it is worth noting that our model proved the potential to work with multi-size images compared to state-of-the-art methods by Luo et al. (2022) and You et al. (2021).

Experimental setup

Datasets

We use two datasets, Break Our Steganographic System Base 1.01 (BOSSbase 1.01) (Bas et al. 2011) and Break Our Watermarking System 2 (BOWS 2) (Mazurczyk and Wendzel 2018) to experiment with the proposed CNN. BOSSbase 1.01 and BOWS have 10,000 Portable Gray Map (PGM) photos of size 512×512 pixels each, and datasets have similarities in the features and the capturing devices used to prevent cover- source mismatch effect (Giboulot et al. 2020). In experimentation, we first establish a baseline for images to use through the following tasks:

(1) Change the size from 512×512 pixels to 256×256 pixels for all images.
(2) Generate stego images from the covers using Spatial UNIversal WAvelet Relative Distortion (S-UNIWARD), and Wavelet Obtained Weights (WOW) steganographic algorithms with 0.2 Bits Per Pixel (*bpp*) and 0.4*bpp*.
(3) Group images into three groups; the first group for training images, the second for validation images, and the third for testing images.
(4) Arrange images into two databases based on which dataset they come from, one for images from BOSSBase 1.01 and another group for photos from a combined set of BOSSBase 1.01 and BOWS 2.
(5) Saving each group of images in the NumPy array improves the system's reading time.

Our datasets are organized into two groups; the first group consists of 10,000 covers, and stego images of BOSSbase 1.01 are split into 4000 training pairs, 1000 validation pairs, and 5000 testing pairs. The second dataset combines images from BOSSBase 1.01 and BOWS 2 to get 20,000 cover/stego pairs. The pairs of the second dataset are split into 14,000 training pairs made from 10,000 pairs of images from BOWS 2 and 4000 pairs of images from BOSSbase 1.01, 1000 validation pairs from BOSSbase 1.01, and 5000 testing pairs from BOSSbase 1.01. For embedding data algorithms, we use Aletheia, an open-source software, to embed the secret data S-UNIWARD and WOW algorithms have been applied with both 0.2*bpp* and 0.4*bpp*.

Moreover, for cross-dataset experiments to verify the training phase stability and the generalization capability of our method to be applicable to JPEG steganography, we also use ALASKA#2 as the third dataset. This experiment is also intended to demonstrate the robustness of our method against overfitting. From the ALASKA#2 dataset (Cogranne et al. 2019), available for steganalysis, we use 15,000 512×512-pixel color images, categorized into three groups: cover images, images with hidden messages using the JPEG version of the universal distortion function (J-UNIWARD) steganographic algorithm, and images with confidential messages using Uniform Embedding Revisited for Difficult Images (UERD) steganographic algorithm. While the cover images are unaltered, the steganographic images have hidden messages that require steganalysis techniques to uncover. Although it poses a challenge for steganalysis, the ALASKA#2 dataset's large and diverse photographic images overcome obstacles in transitioning from research labs to real-world scenarios. To use the ALASKA#2 dataset, images are first converted to PGM format to match BOSSBase 1.01 and BOWS 2, primarily used in our experiments. To verify the effectiveness of ALASKA#2 as a training dataset on our method to predict the images from BOSSBase 1.01, we use S-UNIWARD steganographic algorithm with 0.4 bpp. In our experiment, to identify the stability of the training phase and to prove the robustness of our method against overfitting, we use 80,000 cover-stego images from ALASKA#2 for training, 2000 pairs of cover-stego images from BOSSBase 1.01 for validation, and 5000 pairs of cover-stego images from BOSSBase 1.01 for testing.

Hardware and software resources

We implement our method with Python 3.8.1 and TensorFlow 2. 2. 0 in the windows operating system. The computer features are GeForce RTX 1024 Ti, CUDA version 11.1, AMD 9 3950X 8-Core Processor, and 32 GB of RAM. We also use Google Collaboratory with the environment as Tesla P100 PCIe (16 GB) GPUs, CUDA Version 10. 1, and 25.51 GB of RAM.

Hyper-parameters selection

For this method, we use a batch size of 64; for the network training on a specific payload, we use 100 epochs. The spatial dropout value is 0.1, the momentum of BN momentum is 0.2, the epsilon is set to 0.001, and the norm momentum value is 0.4. The weights for fully connected layers and convolutional layers use a glorot normal initializer, and to regularize the kernels and bias, the L_2 is used. The momentum is set to 0.95 for the stochastic gradient descent optimizer (SGDO), and the learning rate used is 0.001. The slope of the LeakyReLU is set to – 0.1 (Tables 1 and 2).

Table 1 DAC, TPR, and FPR for the existing CNNs and the proposed CNN with dataset BOSSBase 1.01

Algorithm	**S-UNIWARD**						**WOW**					
Payload (bpp)	**0.2**			**0.4**			**0.2**			**0.4**		
CNN	**DAC**	**TPR**	**FPR**	**DAC**	**TPR**	**FPR**	**DAC**	**TPR**	**FPR**	**DAC**	**TPR**	**FPR**
Ye-Net	60.1	–	–	68.7	–	–	66.9	–	–	76.7	–	–
Yedroudj-Net	63.5	–	–	77.4			72.3	–	–	81.1	–	–
Zhu-Net	71.4	–	–	80.5	–	–	76.9	–	–	84.1	–	–
GBRAS-Net	73.6	87.7	29.2	87.1	91.5	17.9	80.3	89.6	19.0	89.8	91.8	14.4
Proposed-Net	79.3	90.5	21.6	93.1	95.5	12.1	90.2	92.1	16.7	94.4	92.5	12.6

Table 2 DAC, TPR, and FPR for the existing CNNs and the proposed CNN with dataset BOSSBase 1.01 combined with BOWS 2

Algorithm	**S-UNIWARD**						**WOW**					
Payload (bpp)	**0.2**			**0.4**			**0.2**			**0.4**		
CNN	**DAC**	**TPR**	**FPR**	**DAC**	**TPR**	**FPR**	**DAC**	**TPR**	**FPR**	**DAC**	**TPR**	**FPR**
Ye-Net	–	–	–	–	–	–	73.6	–	–	–	–	–
Yedroudj-Net	65.6	–	–	–	–	–	75.7	–	–	–	–	–
Zhu-Net	75.7	–	–	83.9	–	–	82.0	–	–	88.2	–	–
GBRAS-Net	77.9	–	–	90.7	–	–	82.7	90.2	16.2	92.4	93.9	12.4
Proposed-Net	83.4	91.0	18.2	97.3	96.0	10.2	92.9	95.4	12.6	97.2	97.6	9.6

Evaluation metrics

To evaluate our method, we mainly consider the Detection ACcuraccy (DAC) obtained by (5), which depends on four classes of the obtained classification results, namely true positive (TP), true negative (TN), false positive (FP), and false negative (FN). In this experiment, TP represents a class of stego images predicted to be stego images, TN represents a class of cover images predicted to be cover images, and FP represents a class of cover images predicted to be stego images. FN represents a class of stego images predicted to be the cover images. Moreover, we consider the Testing ACcuracy (TAC) as presented in Tables 3 and 4, the Training Time (TT) shown in Table 5, False Positive Rate (FPR) got by (6), and the True Positive Rate (TPR) got by (7) are also recorded in Tables 1 and 2.

$$DAC = \left(\frac{TP + TN}{TP + TN + FP + FN} \times 100\right)\% \tag{5}$$

$$FPR = \left(\frac{FP}{FP + TN} \times 100\right)\% \tag{6}$$

$$TPR = \left(\frac{TP}{TP + FN} \times 100\right)\% \tag{7}$$

Experimental results and discussion

To compare the performance of our method to the existing methods, we report results in line with the metrics stated in "Results of a cross-dataset experiment" section to evaluate them comparatively and identify quantitative improvement. We report in Tables 1 and 2 the results in DAC, TPR, and FPR of the considered existing methods and the method we propose in this article. Generally, a significant improvement is identified with our approach in all sizes of payload capacities. However, looking at the results in both tables, it is identified that the higher payload capacities are detected with higher accuracy in all steganalysis methods considered in this article. It also identified that the steganographic payloads are efficiently detected when the dataset is increased because the results achieved within a combination of BOSSBase, and BOWS are better than the ones obtained with only BOSSBase.

Table 3 TAC for the existing CNNs and the proposed CNN with dataset BOSSBase

Algorithm	S-UNIWARD		WOW	
Payload (bpp)	0.2	0.4	0.2	0.4
CNN	TAC	TAC	TAC	TAC
Ye-Net	60.0	68.8	64.9	76.2
Yedroudj-Net	63.3	77.2	72.1	84.1
Zhu-Net	65.7	80.1	76.2	84.4
GBRAS-Net	70.1	81.4	79.3	85.9
Proposed	76.7	86.3	82.1	91.9

Table 4 TAC for the existing CNNs and the proposed CNN with dataset BOSSBase combined with BOWS

Algorithm	S-UNIWARD		WOW	
Payload (bpp)	0.2	0.4	0.2	0.4
CNN	TAC	TAC	TAC	TAC
Ye-Net	–	–	73.9	–
Yedroudj-Net	64.6	–	76.3	–
Zhu-Net	70.7	83.9	80.0	86.2
GBRAS-Net	74.9	83.5	82.6	87.1
Proposed	79.7	84.6	87.1	90.7

Table 5 Approximate training time in minutes for existing steganalysis CNNs and the proposed CNN system

Dataset	BOSSBase1.01		BOSSBase1.01 + BOWS 2	
Payload (bpp)	0.2	0.4	0.2	0.4
CNN				
Ye-Net	80	180	140	280
Yedroudj-Net	100	220	350	400
Zhu-Net	90	195	320	390
GBRAS-Net	300	360	440	540
Proposed-Net	190	221	349	398

The results obtained by Ye et al. (2017) have been improved with our method in a range from 17.7 to 26.3%, which is a significantly high difference. Based on this difference between the results of a technique proposed in Ye-Net, we can conclude that the constant values of the SRM used to calculate the residual maps and the Truncated Linear Unit (TLU) used to enhance the SNR are less performing than the SRM filter banks and the LReLU that we adopt in our CNN.

Comparing our results to the results reported by Yedroudj et al. (2018), it is identified that our CNN outperforms the Ye-Net with a significant superiority in results. Departing from our results and the results with Yedroudj, an improvement ranging between 13.3% and 19.3% is identified. This improvement shows that the 30 SRM filter banks with kernel size 5×5 used in the preprocessing phase, TLU functions used for non-linearity enhancement, BN used to normalize the features in the convolutional layers for feature extraction, and the average pooling are not good as the 30 SRM filter banks with

sizes 5×5 and 3×3 used in the pre-processing phase, and a combination of depthwise separable convolutions with LReLu, and BN functions we used in our CNN.

Moreover, our CNN achieves better results than the CNN proposed Zhang et al. (2020). Using BOSSBase images, the lowest result of the data reported with Zhu-Net has been identified as 13.3% less than the results of our network. This result is yielded with WOW as a steganographic algorithm with the low embedding capacity of our experiment 0.2*bpp*. Considering a dataset obtained by combining BOSSbase and BOWS, which is bigger than the BOSSBase alone, the most significant outperformance percentage of detection accuracy is also identified with WOW as a steganographic algorithm, using a payload capacity of 0.2*bpp*. Our method is 17.2% greater than the Zhu-Net, which shows the efficiency of our approach over the Zhu-Net in detecting the steganographic payload, especially in the low payload capacities.

Additionally, the proposed method improves the detection accuracy in two datasets, notably the BOSSBase 1. 01 and the BOSSBase 1 0.01 combined with BOWS 2. This method works on S-UNIWARD and WOW algorithms with payload capacities 0.2*bpp* and 0.4*bpp*. Considering the results of GBRAS-Net on BOSSBase 1 0.01, the accuracy with the proposed method was improved by 5.7% on S-UNIWARD with 0.2*bpp*, and 6.0% with 0.4*bpp*. Improvements of 9.9% and 4.6% are identified on WOW with 0.2*bpp* and 0.4*bpp*, respectively. The results of GBRAS-Net on BOSSBase 1. 01 combined with BOWS 2, the proposed method improves the accuracy by 5.5% on S-UNIWARD with 0.2*bpp* and 6.69% with 0.4*bpp*. The proposed method also improves the accuracy by 10.2% on WOW with 0.2*bpp* and 4.8% with 0.4*bpp*. Therefore, we can conclude that the approach used to extract and learn the features in our method yields better results in detecting the steganographically altered digital images.

Tables 3 and 4 include the results in the testing stage of the network referred to as TAC in this paper because the testing stage also expresses the behavioural efficiency of a CNN. With the BOSSBase dataset, for the S-UNIWARD algorithm with a payload capacity of 0.2*bpp* the existing methods' TAC has been improved with our approach by 16.7%, 13.4%, 11.0%, and 6.6% for Ye-Net, Yedroudj-Net, Zhu-Net, and GBRAS-net respectively.

With a payload capacity of 0.4*bpp*, our method's TAC is 17.5%, 9.1%, 6.2%, and 4.9% superior to the TAC of Ye-Net, Yedroudj-Net, Zhu-Net, and GBRAS-Net respectively. With WOW steganographic algorithm, our method shows an improvement of the testing accuracy beating the one with S-UNIWARD steganographic algorithm because with a payload capacity of 0.2*bpp* the existing methods' TAC has been improved with our approach by 17.2%, 10.0%, 5.9%, and 2.8% for Ye-Net, Yedroudj-Net, Zhu-Net, and GBRAS-net respectively. With a payload capacity of 0.4*bpp*, our method's TAC is 15.7%, 7.8%, 7.5%, and 6.0% superior to the TAC of Ye-Net, Yedroudj-Net, Zhu-Net, and GBRAS-Net respectively. Based on the improved outperformance of our method, it is identified that our CNN beats the existing CNN in the training stability, which influences the accuracy of the testing phase. However, it is remarked that the proposed method yields better results with the WOW algorithm when compared to S-UNIWARD.

Comparing the results recorded in both Tables 3 and 4, we can also conclude that the proposed method, in the testing phase, performs better with big datasets because in Table 4, which presents the yielded results with a combination of BOSSBase and BOWS, the most remarkable improvement of our method over the existing is 16.7% while the results in Table 3 that show the yielded results with BOSSBase show 17.5% as the most considerable improvement of our method over the existing CNNs. Therefore, it is identified that our CNN achieves better results with more extensive datasets when it is implemented in the same conditions.

Table 5 presents a comparative view of training time for the previously proposed CNNs, and the CNN proposed in our work. To have a valid comparison, we adopted during our experimental setup the hardware and resources used in the previous research (Reinel et al. 2021; Ye et al. 2017; Yedroudj et al. 2018; Zhang et al. 2020). Though the training time cannot be considered an evaluation metric because it is being influenced by several factors, such as hardware-based factors, it is crucial to mention that the proposed steganalysis CNN training time varies between approximately 190 min to around 398 min. The training and validation accuracy curves of the proposed CNN are illustrated in Figs. 5 and 6, where the convergence of our model with the detection of a steganographic payload of 0.2*bpp* is higher than the detection of a steganographic payload of 0.4*bpp*. Figure 5 presents the accuracy curves for BOSSBase 1. 01, and Fig. 6 presents the accuracy curves for a combination of BOSSBase 1. 01 and BOWS 2 data using the WOW algorithm.

The experimental results show that our method improves the detection accuracy in two datasets, notably BOSSBase 1. 01 and BOSSBase 1 0.01 combined with BOWS 2. This method works on S-UNIWARD and WOW algorithms with payload capacities 0.2*bpp* and 0.4*bpp*. Considering the results of recent CNNs on BOSSBase 1 0.01, the performance in detection accuracy was significantly improved for both S-UNIWARD and WOW, with WOW highly enhanced. Figure 7 gives a comparative graph between the existing CNNs and the proposed CNN regarding detection accuracies.

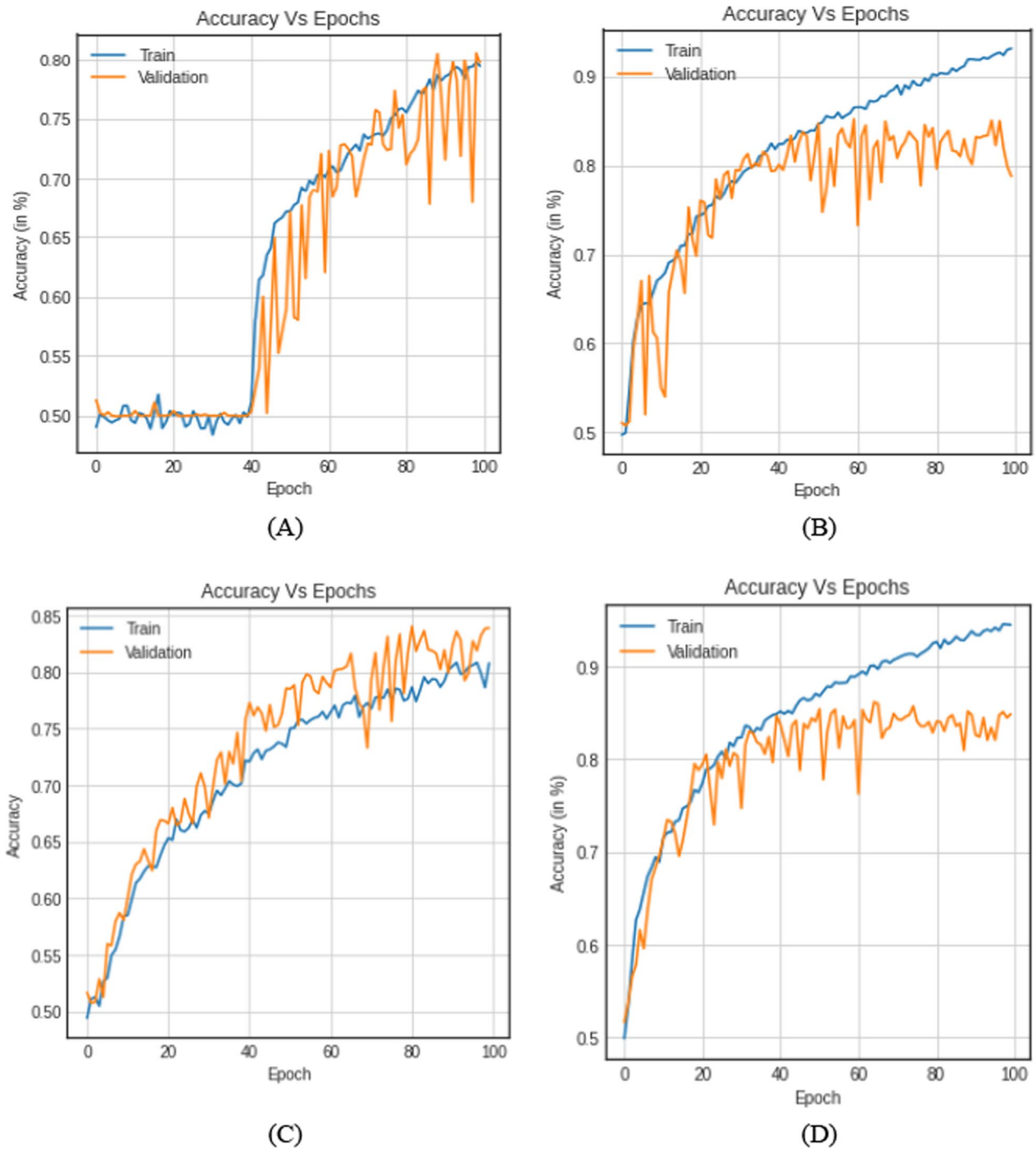

Fig. 5 Accuracy curves for BOSSBase 1. 01 with S-UNIWARD **A** 0.2*bpp*, **B** 0.4*bpp*, and WOW, **C** 0.2*bpp*, **D** 0.4*bpp*

Results comparison with the state-of-the-art methods

To demonstrate the effectiveness of our method, some state-of-the-art techniques are compared to the results of our approach. The considered evaluation metrics for this benchmark are in line with the metrics stated in "Results of a cross-dataset experiment" section to evaluate them comparatively and identify quantitative improvement. We report in Tables 6 and 7 the results in DAC for the considered prior methods and the method we propose in this article.

Table 6 contains our method's obtained results and the existing methods' reported results considering the

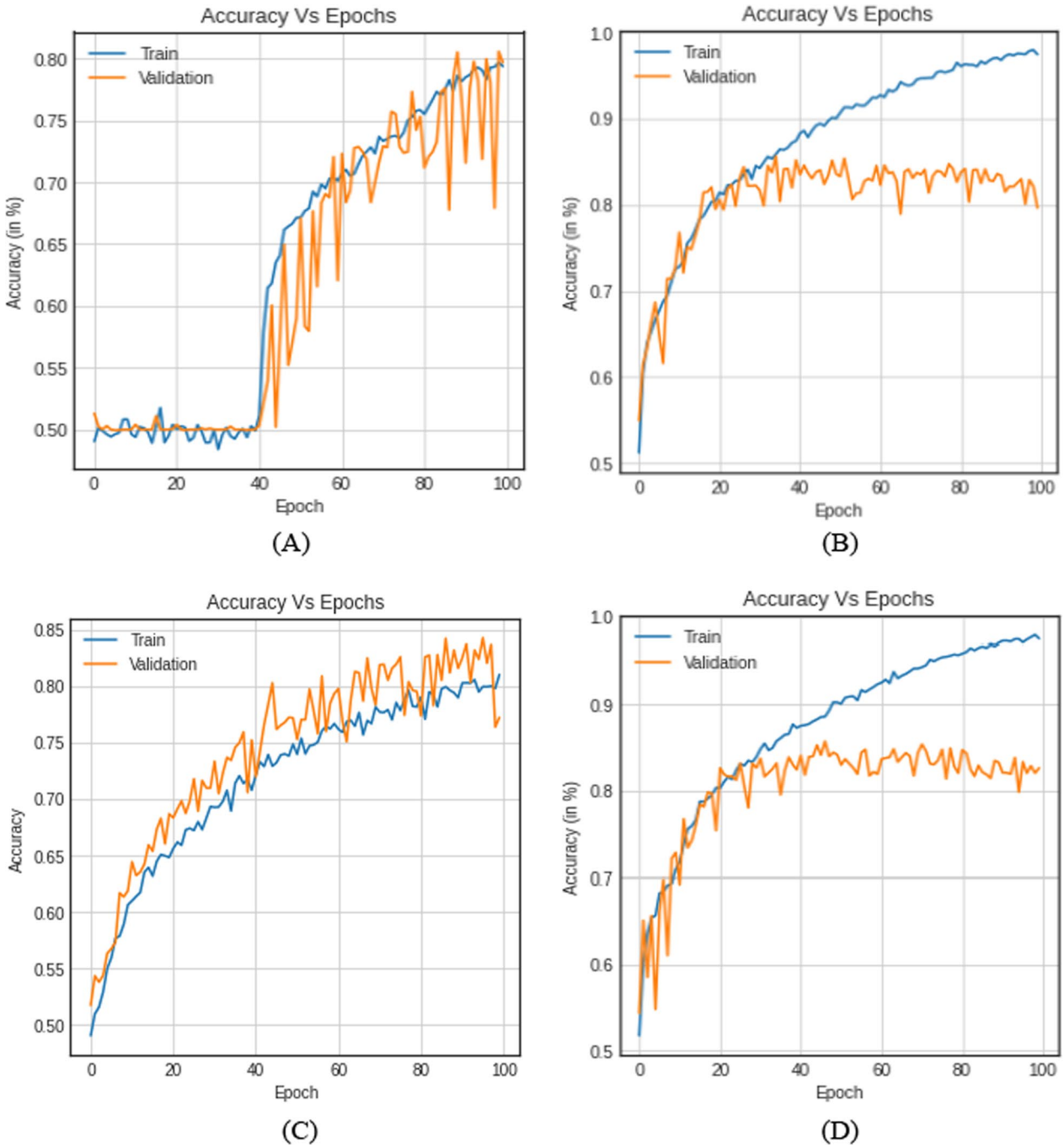

Fig. 6 Accuracy curves for BOSSBase 1.01 + BOWS 2 with S-UNIWARD **A** 0.2*bpp*, **B** 0.4*bpp*, and WOW **C** 0.2*bpp*, **D** 0.4*bpp*

fixed-size benchmark datasets. With these results, we believe WOW, and S-UNIWARD as two experimented steganographic algorithms with payload capacities of 0.2*bpp* and 0.4*bpp* under the BOSSBase 1.01 dataset. Based on the results, it is identified that our method achieves outperforming results in all steganographic algorithms and payload capacities. The reported results show that WOW is more accurately detected than S-UNIWARD. The average percentage raised for the detection accuracy is 7.6%, a significant improvement over the current method (Reinel et al. 2021), and 11.8%, which is also a promising improvement over the proposed method (Zhang et al. 2020).

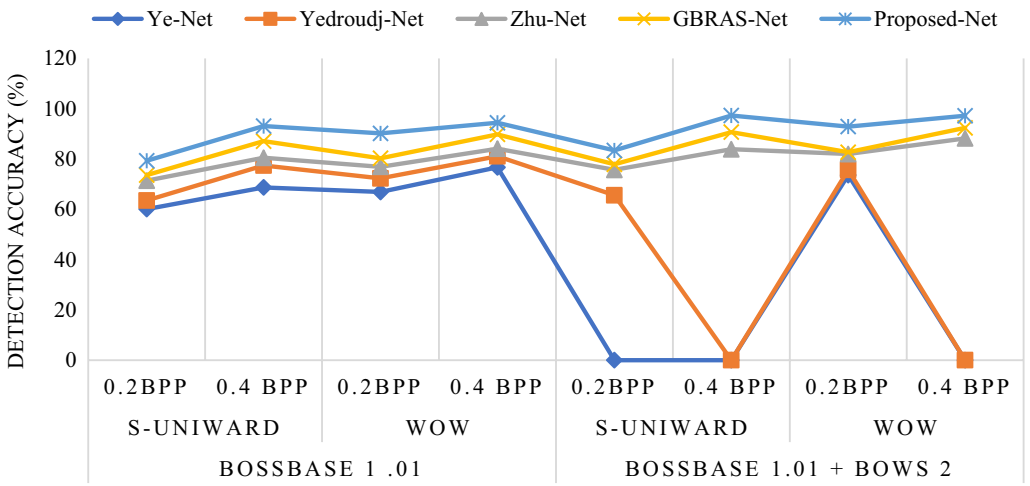

Fig. 7 Comparison between the existing CNNs and the proposed CNN in detection accuracy

Table 6 DAC comparison between our method and the existing methods with fixed-size images

Steganographic algorithm	Proposed		GBRAS		ZHU	
	0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp
WOW	90.2	94.4	80.3	89.8	76.9	84.1
S-UNIWARD	79.3	93.1	73.6	87.1	71.4	80.5

Table 7 DAC comparison between our method and the existing methods with arbitrary-size images under WOW with 0.2*bpp*

Arbitrary size for test	Proposed method	S-CNN	CVT-CNN
512 × 512	77.6	77.2	73.0
512 × 640	73.8	76.8	73.9
640 × 512	75.8	77.0	76.1
640 × 640	77.2	76.4	73.8

Though the problem of steganalysis in images of different sizes is beyond the scope of our work, it is worth noting that our model proved the potential to work with arbitrary-size images compared to the state-of-the-art methods in Luo et al. (2022) and You et al. (2021). To show the potential of our approach in detecting steganography within images of arbitrary size compared to some of the existing works, though we do not deepen the arbitrary-size images treatment, which requires many discussions about the payload, the size, and resolution of images because it is out of the scope of this article, we experiment as of the following.

Departing from the method by You et al. (2021), which proposed a convolutional vision transformer to detect the presence of a steganographic payload as CVT-CNN, and the method by Luo et al. (2022), which presented an end-to-end steganalysis model based on deep learning as S-CNN, we prepare and organize images from BOSS_512 to be used for evaluation of the applicability of our model for training and testing the arbitrary-size images. We convert each image of size 512×512 for the training set from BOSS_512 to an image of arbitrary rectangular size. Considering h and w for the respective height and width of the randomly sized input, with h, w belonging to the range $[3/4 \times 512, 512]$; and the cartesian coordinates of the resulted image's upper corner in the left as (x, y), with x belonging to $[1, 512 - w]$, and y belonging to $[1, 512 - h]$ is generated as well to make sure that host and cropped images fit. The obtained results are recorded in Table 7. Considering the model proposed by You et al. (2021) as a benchmark, we identify an improvement ranging from 0.4 to 0.8% for inquiry images of rectangular shapes with equal height and width. However, our method is inferior in performance compared to S-CNN for rectangular shapes with different sizes in height and width but superior to CVT-CNN in all cases. The outstanding results obtained are highlighted in bold for the models considered in our experiment.

Results of a cross-dataset experiment

In our work, to test our method's generalization capability, we conduct cross-dataset experiments. We use images from the ALASKA#2 dataset to verify the training phase's stability and to assess our method's robustness against overfitting.

Our CNN architecture uses the weights from the training with BOSSBase 1.01 consisting of 4000 cover-stego image pairs. While training our CNN with ALASKA#2, at the 44th epoch, an accuracy of 68% is achieved, 75.8% accuracy is achieved with the validation set of images and 73.2% on the testing set of images. It is worth noting that different from the training set of data from ALASKA#2, the validation and testing data are from BOSSBase 1.01. With the best model obtained at the 4th epoch, our CNN correctly classifies 28,000 cover-stego image pairs from ALASKA#2. Next, we proceed with training our model with 4000 cover-stego image pairs from BOSSBase 1.01, combined with 10,000 cover-stego images pairs from BOWS 2 and 28,000 cover-stego image pairs from ALASKA#2, which make a training set of 42,000 cover-stego images for 38 epochs, with re-initialization of the model's weights departing from those from training the network with 4000 cover-stego image pairs from BOSSbase 1.01, and kept the same sets for validation and training as per the previous experiment. From this experiment, we achieved a training accuracy of 90.5% and a testing accuracy of 93.2% in testing, representing a 3.7% improvement compared to working with BOSSBase 1.01 originated images alone.

Ablation study

To demonstrate the effectiveness of the components of the proposed CNN, Table 8 shows the model's performance comparison between different versions of our system with and without some of the proposed new components, namely the use of 2D depthwise separable convolution instead of traditional convolution as of the recently proposed CNN (Reinel et al. 2021), the use of LeakyReLU as a non-linearity function instead of ReLU recently used in Zhang et al. (2020), and the use of multi-scale pooling instead of global average pooling recently used in Reinel et al. (2021).

Depthwise separable convolution has become increasingly popular in computer vision-based tasks, including Inception and Xception structures, with Xception considered a variant of an Inception module. In this work, the inception entirely separates the correlation between channels, which decreases storage space and enhances the model's expressiveness. To better use the residual information of cover/stego, we create corresponding 2D depthwise separable convolutional blocks, each consisting of a 1×1 convolution and a 3×3 convolution after layer pre-processing in Fig. 4 of the proposed CNN architecture.

This work assumes no dependency between the residuals channel correlation and spatial correlation. To evaluate the efficiency of the used 2D depthwise separable convolutions over the traditional convolutions, we experimented with the CNN architecture we propose in Fig. 4, considering two cases. The first involves training the model with conventional convolutions, and the second with the proposed 2D depthwise separable convolutions. In both cases, we consider WOW and S-UNIWARD steganographic algorithms with 0.2*bpp* and 0.4*bpp* payload capacities. Based on the results yielded in Table 8, the model with 2D depthwise separable convolutions improves the results in a percentage ranging from 9.6 and 4.2; even when the LReLU is not used but keeping the 2D depthwise separable convolutional layers, the results obtained are always better than the results with the traditional convolutional layers.

LReLU function is used in all blocks to avoid vanishing gradients, permit backward communication, and keep

Table 8 Detection accuracy comparison with and without some components to measure the impact of each component on the system performance

Architecture	Detection accuracy			
	WOW		S-UNIWARD	
	0.2 bpp	0.4 bpp	0.2 bpp	0.4 bpp
Proposed method	90.2	94.4	79.3	93.1
Proposed without 2D depthwise	81.6	90.2	74.9	88.2
Proposed without LReLu	83.4	91.7	77.4	90.1
Proposed without Multi-scale pooling	79.2	87.9	74.1	86.2
Proposed without 2D depthwise and LReLu	79.8	88.3	75.5	87.4
Proposed without 2D depthwise and multi-scale pooling	77.1	85.6	73.2	84.7
Proposed without LReLu and Multi-scale pooling	78.2	87.0	72.4	86.3

the model weights positive. Using the LReLu, neurons selectively react to crucial input signals; hence we achieve significant efficiency with features. To identify the contribution of the proposed LReLU, we train our model in Fig. 4, first with ReLu, and second with LReLu in all blocks. In both cases, we train with WOW and S-UNIWARD with a steganographic payload of 0.2*bpp* and 0.4*bpp*. Departing from the results reported in Table 8, it is proved that the proposed model with LReLU outperforms the model with ReLU with an average accuracy of 3.6% which is significant in steganalysis experiments.

To identify the significance of the multi-scale pooling operation to improve the results, we performed experiments in two scenarios by training our model with WOW and S-UNIWARD as steganographic algorithms with 0.2 bpp and 0.4 bpp as payload capacities. In the first scenario, we experimented with our CNN by replacing the multi-scaling pooling operation with the global average pooling operation. In the second scenario, we added the spatial pooling module for multi-scaling operation in the first layer for the classification block of the proposed CNN. As reported in Table 8, the experimental results proved the outperformance of the proposed multi-scale pooling in all tested cases. The average improvement of the detection accuracy is 7.4%, which shows a significant impact of using this type of pooling layer.

Conclusion

Research to improve the steganalysis performance has been carried out and proved that CNN achieves outstanding results over the standard ML-based handcraft features. This article focuses on the CNN paradigm to design a new CNN for spatial domain image steganalysis. Our method outperforms the existing techniques in detection accuracy and training stability based on the cross-dataset experiments results achieved. Our contributions to the existing CNNs are: (1) using 2D depthwise separable convolutions to prevent kernels skipping for channel correlation and spatial correlation of the residuals to enhance the training phase and prevent the overfitting issue. (2) using LReLu for non-linearity to avoid the vanishing gradients to make the backward communication possible and enhance the network's convergence, and (3) using multi-scale pooling to compact the feature maps regardless of the previous feature maps sizes and keep the spatial information of various sizes. For experiments, we use two datasets: BOSSBase 1.01 and a combination of BOSSBase 1.01 with BOWS 2. We also apply S-UNIWARD and WOW as steganographic algorithms for random data embedding. We also use the ALASKA#2 dataset with WOW and S-UNIWARD algorithms to verify the training phase stability. The experimental results demonstrate a significant outperformance of our method over the existing methods to accurately detect the steganographic payload for both fixed-size and arbitrary-size images.

For future works, we aim to apply this proposed method with other datasets, such as agricultural and medical images, to study and analyze the behaviour of this CNN in other image classification problems. Moreover, by improving this same CNN, we aim to identify the exactly altered pixels of the stego image by combining some features of this work with the methods proposed Chen et al. (2020), Sun et al. (2019), and Yang et al. (2019).

Acknowledgements
The authors would like to thank all the lab and research group members.

Author contributions
NJDLC: conceptualization, methodology, software, formal analysis, investigation, writing original draft, visualization. TA: conceptualization, methodology, writing review and editing, supervision, project administration, funding acquisition. All authors read and approved the final manuscript.

Funding
This research was supported by the Ministry of Education, Culture, Research and Technology, The Republic of Indonesia, and Institut Teknologi Sepuluh Nopember.

Availability of data and materials
https://drive.google.com/drive/folders/1jkr01hjH3YFQcQNociiAxnp1Zgn6jYJX.

Declarations

Competing interests
All authors have no competing interests.

Received: 4 February 2023 Accepted: 29 March 2023
Published online: 02 September 2023

References

Ahmad T, Fatman AN (2022) Improving the performance of the histogram-based data hiding method in the video environment. J King Saud Univ Comput Inf Sci 34(4):1362–1372. https://doi.org/10.1016/j.jksuci.2020.04.013

Alsabhany AA, Ali AH, Ridzuan F, Azni AH, Mokhtar MR (2020) Digital audio steganography: systematic review, classification, and analysis of the current state of the art. Comput Sci Rev. https://doi.org/10.1016/j.cosrev.2020.100316

Bas P, Filler T, Pevný T (2011) "Break our steganographic system": the ins and outs of organizing BOSS. In: LNCS (vol 6958). Springer-Verlag, Berlin

Chen Y, Tao J, Liu L, Xiong J, Xia R, Xie J, Zhang Q, Yang K (2020) Research of improving semantic image segmentation based on a feature fusion model. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-020-02066-z

Cogranne R, Giboulot Q, Bas P (2019) Documentation of alaskav2 dataset scripts: A hint moving towards steganography and steganalysis into the wild. Available from https://alaska.utt.fr/

De La Croix NJ, Islamy CC, Ahmad T (2022a) Secret message protection using fuzzy logic and difference expansion in digital images. In: 2022a IEEE Nigeria 4th international conference on disruptive technologies for sustainable development (NIGERCON), pp 1–5. https://doi.org/10.1109/NIGERCON54645.2022.9803151

De La Croix NJ, Islamy CC, Ahmad T (2022b) Reversible data hiding using pixel-value-ordering and difference expansion in digital images. In: 2022b

IEEE international conference on communication, networks, and satellite (COMNETSAT), pp 33–38. https://doi.org/10.1109/COMNETSAT56033.2022.9994516
Ferreira WD, Ferreira CBR, da Cruz Júnior G, Soares F (2020) A review of digital image forensics. Comput Electr Eng. https://doi.org/10.1016/j.compeleceng.2020.106685
Giboulot Q, Cogranne R, Borghys D, Bas P (2020) Signal processing: image communication effects and solutions of cover-source mismatch in image steganalysis. Signal Process Image Commun. https://www.sciencedirect.com/science/article/pii/S0923596520300941
Guttikonda JB, Sridevi R (2019) A new steganalysis approach with efficient feature selection and classification algorithms for identifying the stego images. Multimed Tools Appl 78(15):21113–21131. https://doi.org/10.1007/s11042-019-7168-5
Hussain I, Zeng J, Qin X, Tan S (2020) A survey on deep convolutional neural networks for image steganography and steganalysis. KSII Trans Internet Inf Syst 14(3):1228–1248. https://doi.org/10.3837/tiis.2020.03.017
Ikhlayel M, Hariadi M, Ketut I, Pumama E (2019) Copy-move forgery detection based on modified multi-scale feature extraction and CMFD-SIFT. In: IJCSNS international journal of computer science and network security (vol 19, Issue 6).
Liu J, Lu W, Zhan Y, Chen J, Xu Z, Li R (2020) Efficient binary image steganalysis based on ensemble neural network of multi-module. J Real-Time Image Proc 17(1):137–147. https://doi.org/10.1007/s11554-019-00885-8
Lopez-Hernandez J, Martinez-Noriega R, Nakano-Miyatake M, Yamaguchi K (2008) Detection of BPCS-steganography using SMWCF steganalysis and SVM. In: International Symposium on Information Theory and Its Applications, pp. 1–5. https://doi.org/10.1109/ISITA.2008.4895497
Luo G, Wei P, Zhu S, Zhang X, Qian Z, Li S (2022) Image steganalysis with convolutional vision transformer. In: ICASSP, IEEE international conference on acoustics, speech and signal processing—proceedings, 2022-May, pp 3089–3093. https://doi.org/10.1109/ICASSP43922.2022.9747091
Mazurczyk W, Wendzel S (2018) Information hiding: challenges for forensic experts. In: Communications of the ACM (vol 61, Issue 1, pp 86–94). Association for computing machinery. https://doi.org/10.1145/3158416
Nissar A, Mir AH (2010) Classification of steganalysis techniques: a study. Digit Signal Process Rev J 20(6):1758–1770. https://doi.org/10.1016/j.dsp.2010.02.003
Płachta M, Krzemień M, Szczypiorski K, Janicki A (2022) Detection of image steganography using deep learning and ensemble classifiers. Electronics. https://doi.org/10.3390/electronics11101565
Prayogi IB, Ahmad T, de La Croix NJ, Maniriho P (2021) Hiding messages in audio using modulus operation and simple partition. In: Proceedings of 2021 13th international conference on information and communication technology and system, ICTS 2021, pp 51–55. https://doi.org/10.1109/ICTS52701.2021.9609028
Rahman CR, Arko PS, Ali ME, Iqbal Khan MA, Apon SH, Nowrin F, Wasif A (2020) Identification and recognition of rice diseases and pests using convolutional neural networks. Biosys Eng 194:112–120. https://doi.org/10.1016/j.biosystemseng.2020.03.020
Reinel TS, Brayan AAH, Alejandro BOM, Alejandro MR, Daniel AG, Alejandro AGJ, Buenaventura BJA, Simon OA, Gustavo I, Raul RP (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
Selvaraj A, Ezhilarasan A, Wellington SLJ, Sam AR (2021) Digital image steganalysis: a survey on the paradigm shift from machine learning to deep learning-based techniques. IET Image Proc 15(2):504–522. https://doi.org/10.1049/ipr2.12043
Shankara DD, Upadhyay PK (2019) Blind steganalysis for JPEG images using SVM and SVM-PSO classifiers. Int J Innov Technol Explor Eng 8(11):1239–1246. https://doi.org/10.35940/ijitee.K1250.09811S19
Shehab DA, Alhaddad MJ (2022) Comprehensive survey of multimedia steganalysis: techniques, evaluations, and trends in future research. Symmetry. https://doi.org/10.3390/sym14010117
Sun Y, Zhang H, Zhang T, Wang R (2019) Deep neural networks for efficient steganographic payload location. J Real-Time Image Proc 16(3):635–647. https://doi.org/10.1007/s11554-019-00849-y
Tabares-Soto R, Ramos-Pollán R, Isaza G, Orozco-Arias S, Ortíz MAB, Arteaga HBA, Rubio AM, Grisales JAA (2020) Digital media steganalysis. In: Digital media steganography: principles, algorithms, and advances (pp 259–293). Elsevier, Amsterdam. https://doi.org/10.1016/B978-0-12-819438-6.00020-7
Tabares-Soto R, Arteaga-Arteaga HB, Mora-Rubio A, Bravo-Ortíz MA, Arias-Garzón D, Alzate Grisales JA, Burbano Jacome A, Orozco-Arias S, Isaza G, Ramos Pollan R (2021) Strategy to improve the accuracy of convolutional neural network architectures applied to digital image steganalysis in the spatial domain. PeerJ Comput Sci 7:e451. https://doi.org/10.7717/peerj-cs.451
Tereshchenko SN, Perov AA, Osipov AL (2021) Features of Applying Pretrained Convolutional Neural Networks to Graphic Image Steganalysis. Optoelectron Instrum and Data Processing 57(4):419–425. https://doi.org/10.3103/S8756699021040117
Xiang L, Guo G, Yu J, Sheng SV, Yang P (2020) A convolutional neural network-based linguistic steganalysis for synonym substitution steganography. Math Biosci Eng 17(2):1041–1058. https://doi.org/10.3934/mbe.2020055
Yang C, Liu F, Ge S, Lu J, Huang J (2019) Locating secret messages based on quantitative steganalysis. Math Biosci Eng 16(5):4908–4922. https://doi.org/10.3934/mbe.2019247
Ye J, Ni J, Yi Y (2017) Deep learning hierarchical representations for image steganalysis. IEEE Trans Inf Forensics Secur 12(11):2545–2557. https://doi.org/10.1109/TIFS.2017.2710946
Yedroudj M, Comby F, Chaumont M (2018) Yedrouj-Net: An efficient CNN for spatial steganalysis. http://arxiv.org/abs/1803.00407
You W, Zhang H, Zhao X (2021) A siamese CNN for image steganalysis. IEEE Trans Inf Forensics Secur 16:291–306. https://doi.org/10.1109/TIFS.2020.3013204
Zhang R, Zhu F, Liu J, Liu G (2020) Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis. IEEE Trans Inf Forensics Secur 15:1138–1150. https://doi.org/10.1109/TIFS.2019.2936913

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.