Neural Processing Letters (2022) 54:853–870
https://doi.org/10.1007/s11063-021-10661-0

MixNet: A Robust Mixture of Convolutional Neural Networks as Feature Extractors to Detect Stego Images Created by Content-Adaptive Steganography

E. Amrutha[1] · S. Arivazhagan[1] · W. Sylvia Lilly Jebarani[1]

Accepted: 9 October 2021 / Published online: 1 November 2021

Abstract

Digital steganography, the modern version of the ancient art of data hiding is a prevalent tool for covert communication. Steganalysis, at least as old as steganography comes handy to unearth such hidden channels. The illegal act of information hiding through steganography of digital images can be overcome effectively only by using intelligent steganalytic techniques. In this paper, a novel MixNet framework comprising of six Convolutional Neural Networks (CNNs) is proposed as feature extractors for accomplishing generic steganalysis of spatial content-adaptive algorithms with better detection accuracy. Since the spatial content-adaptive algorithms embed secret bits in the hard to model components of the image like edges or textures, inputs to the CNNs are initially filtered using high pass filters to obtain the embedded content in the form of noise residual. Hierarchical features extracted from these networks are then concatenated and used to train Support Vector Machine classifier. Experimentation is performed using the benchmark BOSSbase v1.01 cover images and stego images are created with three state-of-the-art algorithms HUGO-BD, S-UNIWARD and WOW at five relative payloads 0.1–0.5 bits per pixel (bpp). The experimental results show that the proposed MixNet outperforms the compared related works in literature and proves the robustness of MixNet in detecting content-adaptive steganography.

Keywords Content-adaptive steganography · Convolutional neural network · Noise residual · Support vector machine classifier

1 Introduction

Digital steganography, the modern version of the ancient art of data hiding is a prevalent tool for covert communication. Steganography hides a secret message that can be of any multimedia type such as text, image or audio within an innocuous multimedia data (i.e., cover) in such a way that changes are undetectable. The counterpart of steganography is steganalysis.

✉ E. Amrutha
amrutha@mepcoeng.ac.in

[1] Centre for Image Processing and Pattern Recognition, Department of Electronics and Communication Engineering, MEPCO Schlenk Engineering College (Autonomous), Sivakasi 626005, India

The other forms of information hiding are cryptography and watermarking. Cryptography involves coding the message using an encryption key and sending it as ciphertext in scrambled form [1]. Watermarking hides a secret pattern or image inside a host (cover) image to convey a copyrighted content or ownership. Prevalent choice of cover medium is a digital image due to its ability to conceal a payload with invisible effects [2].

Fundamentally, image steganography can be attacked by a steganalyst in two ways namely, passive or active. In passive or generic steganalysis, the aim is to identify whether an image carries a payload or not, whereas in active steganalysis the specific algorithm which was used to embed the payload has to be identified. Generic steganalysis itself defeats the main goal of steganography (concealing secret) by identifying the suspicious stego image [3]. This will aid active steganalysis which needs to go a long way for extraction of secret. Steganographic algorithms that select image contents like edges of image for hiding payload in raw pixels are referred to spatial content-adaptive steganographic algorithms [4]. Since these algorithms leave minimal traces of hidden information, it is necessary to extract image-content independent features in order for the steganalyst to achieve generic steganalysis. This paper presents a robust generic steganalysis framework by framing modern convolutional neural networks to extract highly sophisticated content independent features and classification of those features by using support vector machine classifier to discriminate stego images from innocent cover images.

The rest of the paper is organized as follows: Sect. 2 presents a brief review of literature, Sect. 3 elaborates the proposed methodology, Sect. 4 provides the experiment settings along with dataset used for experiments, Sect. 5 enumerates the experimental results and discussion and finally Sect. 6 concludes the presented work.

2 Review of Literature

Over the past two decades, many approaches of steganalysis have been proposed [5–7]). Traditionally, steganalysis is endeavored using two stages. The first stage is extraction of handcrafted features which must be able to model the embedding distortions in the image caused by any steganographic algorithm. The second stage is classification using binary Support Vector Machine (SVM) or ensemble of classifiers to identify cover and stego images carrying hidden secret. Many researchers have developed high dimensional feature sets and powerful classifiers to detect steganography.

In [8], authors developed a universal steganalyzer called Spatial Rich Model (SRM) composed of multiple rich image residual sub-models. The sub-models take into consideration various types of neighboring samples relationships of noise residuals obtained by linear and non-linear filters. Their framework significantly improved detection rates for two adaptive methods Highly Undetectable stegGO [9] (HUGO) and Edge Adaptive (EA) algorithm and one non-adaptive ± embedding algorithm. [10] developed a low dimensional generic steganalyzer based on spatial and transform domain features to detect stego images with low volume payloads.

The authors in [11] stated the limitation in existing adaptive steganography i.e., it is possible for steganalyzers to estimate the embedded regions from an image. By making use of embedding probabilities of pixels, they proposed an adaptive steganalytic scheme by designing a feature extraction process that would assign higher weights to pixels with high embedding probabilities and vice versa. The experiments were carried out on Wavelet Obtained Weights [12] (WOW), Highly Undetectable steGO Gibbs construction with Bounding Distortion [13] (HUGO-BD), Spatial Version of the UNIversal Wavelet Relative Distortion [14] (S-UNIWARD), and EA algorithms. They achieved remarkable results compared to original Spatial Rich Model (SRM). Yu et al. [15] proposed a spatial steganalytic scheme which is based on redistributed residuals and developed a diverse ensemble classifier.

A non-linear feature map based steganalysis method for spatial content-adaptive steganographic algorithms for grayscale and color images is proposed in [16]. In this approach the authors proposed a non-linear feature transformation on maxSRMd2 features [17] for grayscale images and Spatio-Color Rich Model (SCRM) features for color images using Nyström approximation on various types of kernels. They used a simple linear classifier to mark non-linear boundary between cover and stego classes. Their method not only achieved improvement in detection accuracy of binary classifiers but also worked well as quantitative detectors.

An improved version of texture feature i.e., Local Binary Pattern (LBP) called as threshold LBP (TLBP) is proposed in [18]. In this work, a set of high-order derivative filters is used to obtain residual images and then the TLBP operation is performed on them. From the TLBP images, second order co-occurrence matrix features are formed by feature aggregation process similar to the co-occurrence symmetrization in SRM. This method prevailed successfully over SRM features for state-of-the-art content-adaptive steganographic algorithms.

The difficulty of steganalysis in images due to edges and texture information is addressed in [19]. The authors proposed to pre-classify the cover source using k-means algorithm for improving the performance of multiple existing features. Feature separability analysis of SRM and TLBP features using Fisher score of these features is performed in [20] to improve the detection accuracy of both spatial and transform domain content-adaptive steganography.

In these traditional techniques, feature extraction process is carried out with human supervision and in order for obtaining good accuracy in classification, the steganalyst must have very good domain knowledge about steganography and steganalysis. This is a time-consuming process, that is to learn the fundamentals of steganographic algorithm available and identifying the pattern in which the payload is embedded and so on.

Inspired by the success of deep learning architectures for visual recognition tasks, Convolutional Neural Networks (CNNs) have embarked a strong footmark into steganalysis domain also. The first CNN architecture for steganalysis was developed by Tan and Li [21]. The authors implemented a Stacked Convolutional Auto-Encoder to capture statistical regularities in each layer to model cover and stego images. After their approach, notable CNN based steganalyzers were proposed by authors in [22–26]. With the introduction of deep residual networks [27], the authors established a deeper convolution layer architecture for steganalysis. In this work, reduction in detection accuracy is achieved with multiple layers of residual blocks that help to learn discriminating features. Wu et al. introduced adaptive image content suppression CNN in [28]. This network preserved the subtle stego noise throughout the network and achieved better detection accuracy compared to existing CNN models. End-to-end learnable CNN models are proposed by Wang et al. in [29] and Boroumand et al. in [30]. The Steganalysis Residual Network (SRNet) proposed in [30] is a deep architecture with more than eighty layers of convolution to capture the imperceptible difference between cover and stego images. In [31], Zhang et al. introduced a multi-scale CNN model with residual blocks for spatial steganalysis.

A targeted steganalysis approach to detect S-UNIWARD algorithm is proposed by Kim et al. [32] using a dual channel CNN and dual network CNN based steganalysis. In this approach additional data is first embedded to any given image using S-UNIWARD algorithm and the original along with the difference between original and embedded images are fed as input to CNN. Ren et al. [33] proposed a Learned Selection-Channel-Aware (LSCA) deep learning steganalytic architecture which learns from the embedding probability map of input image along with original image. The base architecture is taken from XuNet and YeNet [34]. A ChAnneLPruning-Assisted (CALPA) deep residual network architecture search approach is proposed by Tan et al. [35] as an effort to shrink the existing vast CNN architectures by. They have used the existing SRNet [30] and XuNet [23] models to form a baseline network and adopted a channel pruning methodology from ThiNet [36] for every convolution layer to reduce the number of channels.

Zhang et al. [37] proposed an enhanced architecture for steganalysis by replacing normal convolution layers by depth-wise separable convolution blocks and spatial pyramid pooling in the place of normal pooling. Liu et al. [38] proposed a new CNN with diverse filter modules (DFMs) and squeeze-and-excitation modules (SEMs) to learn the embedding artefacts made by WOW and S-UNIWARD algorithms. In this architecture, input is preprocessed with 30 HPFs from SRM, and then passed on to three layers of DFMs and SEMs and finally classified using a fully-connected layer with softmax activation.

Though there are tremendous CNN based steganalyzers [39, 40], usage of CNN as a feature extractor is hardly introduced. The detection error rates are still low for low volume payloads and also existing architectures have increased complexity in terms of number of layers. Such downsides of modern deep learning based architectures and traditional steganalytic techniques that are mentioned earlier are addressed in this paper. The major contributions of the research work presented in this paper are highlighted as follows:

- A novel MixNet steganalytic framework comprised of six convolutional neural networks as feature extractors to extract unique and discriminating features to differentiate cover images and stego images created by content-adaptive steganography.
- Usage of diverse filters in the pre-processing layer of the CNNs to bring out the subtle stego noise content residing in the image contents in the form of noise residues.
- Concatenation of features extracted from noise residues using the proposed multiple CNNs proves to be a robust feature set for steganalysis.
- The novel MixNet proposed in this paper achieves passive steganalysis of cover and stego images with various payloads with high accuracy.

3 Proposed MixNet Steganalysis Framework

The proposed MixNet Steganalysis framework is instigated in two stages. In Stage 1, six CNN architectures are trained to discriminate between cover and stego images. In Stage 2, the trained CNN architectures are used as feature extractors to extract content-independent features and these features are fed to Support Vector Machine Classifier to achieve generic steganalysis with reduced error rate.

3.1 Stage 1: Convolutional Neural Network Architectures

In this research work, CNN is proposed specially for detecting content-adaptive steganographic algorithms to improve statistical modelling. It has three advantages. Firstly, CNN can learn features automatically based on the pixel distribution. Secondly, the convolution operation in CNN captures dependencies among pixels, which is a key factor for steganalysis. Thirdly, the parameter sharing mode in CNN significantly reduces the number of trainable parameters and enables CNN to deal with large sized images. In this work, two CNN architectures inspired from XuNet [23] and QianNet [24] are considered as base CNNs and some modification to those architectures are carried out in order to get improved architectures for detecting content-adaptive steganographic algorithms. Figure 1 shows the two base CNN architectures used in this work which are to be used as feature extractors for Stage 2 and three different types of preprocessing filters.

In the convolution module of the base architecture 2 shown in Fig. 1b, a new non linear activation function referred as Gaussian activation [24] is used in the first two convolution modules. The gaussian activation function of input x is expressed in Eq. (1), where σ is the spread of the gaussian curve.

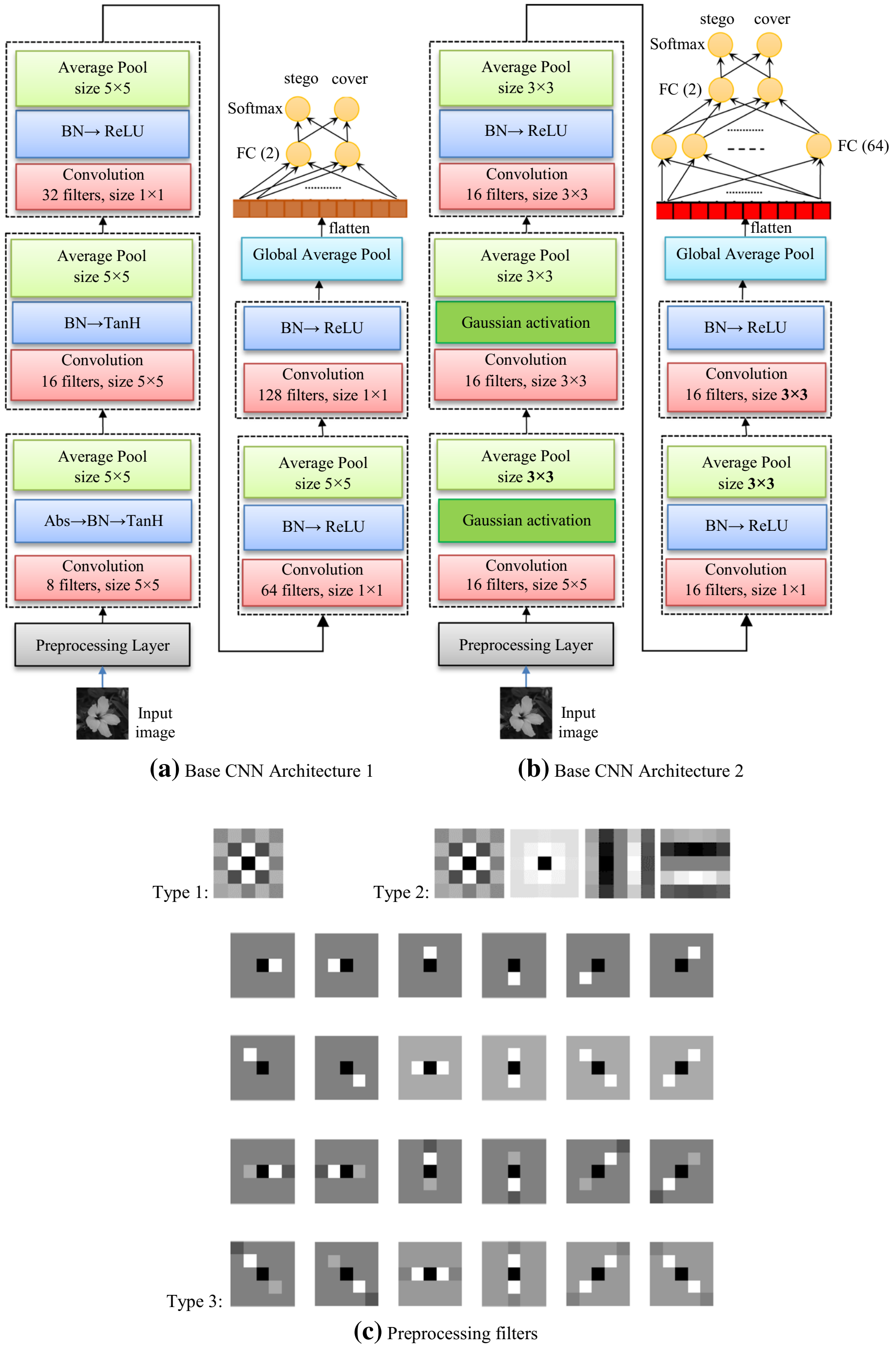
$$\boldsymbol{f}(\boldsymbol{x}) = 1 - \exp\left(-\frac{x^2}{\sigma^2}\right) \tag{1}$$

For the first two convolution layers BN is disabled and for next three convolution layers BN is used followed by ReLU activation. This is done to study the Gaussian activation function. After the global averaging layer, compared to architecture 1, one additional FC layer with 64 neurons is added. Finally, there is a FC layer with 2 neurons and Softmax activation for classification. Tables 1, 2 give the details of the proposed base CNN architecture 1 and 2, respectively. As given in Fig. 1c, the first preprocessing filter type is a high pass filter (KV). Since the content-adaptive steganographic algorithms embed secret bits in the high frequency components of the image i.e., edges and textures, filtering using a high pass filter will help to obtain the embedded content in the form of noise residual. The second type of preprocessing filters are two pairs of high pass filters and Gabor filters. Gabor filters are special classes of band pass filters that are well established for texture analysis. The third type of preprocessing filter is a group of twenty four linear Spatial Rich Model (SRM) filters (high pass filters) that are designed to capture diverse dependencies among neighboring pixels. Note that, the images are filtered by these filters in the preprocessing layer before passing through the convolution layers. As identified in [31], adding a preprocessing stage with many kinds of high-pass filters tend to yield better accuracy rather using images directly or with single filter. With this motivation, the three types of preprocessing filters are used for the base CNN architectures to form six different CNN types. The formation of CNN are as follows:

- CNN Type I—Base CNN Architecture 1 with Type 1 preprocessing filters
- CNN Type II—Base CNN Architecture 1 with Type 2 preprocessing filters
- CNN Type III—Base CNN Architecture 1 with Type 3 preprocessing filters
- CNN Type IV—Base CNN Architecture 2 with Type 1 preprocessing filters
- CNN Type V—Base CNN Architecture 2 with Type 2 preprocessing filters
- CNN Type VI—Base CNN Architecture 2 with Type 3 preprocessing filters

The idea behind framing these six architectures is to utilize the different types of preprocessing filters to filter the images and combining the features learned from all these vibrant mix of convolution layers of six CNN types for the ultimate goal of achieving generic steganalysis with high detection accuracy. Since each convolution layer in these six types of CNN models, processes the input image data with a rich set of filters that are learned during training procedure, it adds an additional level of image content suppression and on the whole, image content independent features are extracted with the help of concatenating all the features from the MixNet.

These six architectures are individually trained to achieve generic steganalysis. Momentum based Stochastic Gradient Descent backpropagation algorithm is used for training the convolutional neural networks. Categorical cross-entropy loss or logarithmic loss is used for computing the difference in predicted probability and the actual class value (0—cover or



(a) Base CNN Architecture 1

(b) Base CNN Architecture 2

(c) Preprocessing filters

Fig. 1 Schematic diagram of the proposed convolutional neural network architectures and preprocessing filters

Table 1 Parameters of the proposed base CNN architecture 1

Layer	Function	Kernel depth, size	S, P	Output feature map, size	Learnable Parameters		
					Weights	Bias	Total
Input	–	–	–	1, 512 × 512	–	–	–
Group 1	Conv	8, 5 × 5	1,0	8, 508 × 508	5 × 5 × 1 × 8	1 × 1 × 8	208
	Abs, BN, Tanh	–	–	8, 508 × 508	–	–	–
	Average pool	– , 5 × 5	2,0	8, 252 × 252	–	–	–
Group 2	Conv	16, 5 × 5	1,0	16, 248 × 248	5 × 5 × 8 × 16	1 × 1 × 16	3216
	BN, Tanh	–	–	16, 248 × 248	–	–	–
	Average pool	– , 5 × 5	2,0	16, 122 × 122	–	–	–
Group 3	Conv	32, 1 × 1	1,0	32, 122 × 122	1 × 1 × 16 × 32	1 × 1 × 32	544
	BN, ReLU	–	–	32, 122 × 122	–	–	–
	Average pool	– , 5 × 5	2,0	32, 59 × 59	–	–	–
Group 4	Conv	64, 1 × 1	1,0	64, 59 × 59	1 × 1 × 32 × 64	1 × 1 × 64	2112
	BN, ReLU	–	–	64, 59 × 59	–	–	–
	Average pool	– , 5 × 5	2,0	64, 28 × 28	–	–	–
Group 5	Conv	128, 1 × 1	1,0	128, 28 × 28	1 × 1 × 64 × 128	1 × 1 × 128	8320
	BN, ReLU	–	–	128, 28 × 28	–	–	–
	Global average pool	–	–	128, 1 × 1	–	–	–
6	FC, ReLU (2)	–	–	2, 1 × 1	2 × 128	2 × 1	258
Output	FC (Softmax)	–	–	2, 1 × 1	–	–	-

Table 2 Parameters of the proposed base CNN architecture 2

Layer	Function	Kernel depth, size	S, P	Output feature map, size	Learnable parameters		
					Weights	Bias	Total
Input	–	–	–	1, 512 × 512	–	–	–
Group 1	Conv	16, 5 × 5	1,0	16, 508 × 508	5 × 5 × 1 × 16	1 × 1 × 16	416
	Gaussian	–	–	16, 508 × 508	–	–	–
	Average pool	– , 3 × 3	2,0	16, 253 × 253	–	–	–
Group 2	Conv	16, 3 × 3	1,0	16, 251 × 251	3 × 3 × 16 × 16	1 × 1 × 16	2320
	Gaussian	–	–	16, 251 × 251	–	–	–
	Average pool	– , 3 × 3	2,0	16, 125 × 125	–	–	–
Group 3	Conv	16, 3 × 3	1,0	16, 123 × 123	3 × 3 × 16 × 16	1 × 1 × 16	2320
	BN, ReLU	–	–	16, 123 × 123	–	–	–
	Average pool	– , 3 × 3	2,0	16, 61 × 61	–	–	–
Group 4	Conv	16, 3 × 3	1,0	16, 59 × 59	3 × 3 × 16 × 16	1 × 1 × 16	2320
	BN, ReLU	–	–	16, 59 × 59	–	–	–
	Average pool	– , 3 × 3	2,0	16, 29 × 29	–	–	–
Group 5	Conv	16, 3 × 3	1,0	16, 27 × 27	3 × 3 × 16 × 16	1 × 1 × 16	2320
	BN, ReLU	–	–	16, 27 × 27	–	–	–
	Global Average pool	–	–	16,1 × 1	–	–	–
6	FC, ReLU (64)	–	–	64, 1 × 1	64 × 16	64 × 1	1088
7	FC, ReLU (2)	–	–	2, 1 × 1	2 × 64	2 × 1	130
Output	FC (Softmax)	–	–	2, 1 × 1	–	–	–

1—stego). Since this loss function penalizes the score based on the distance between predicted probability's distance from the actual class value, a small score of 0.1 or 0.2 refers to small differences and a high score of 0.9 or 1.0 refers to large difference. When this loss is minimized while training and testing, a better model is finally obtained and this helps to predict the cover and stego images with high degree of confidence. The pseudo algorithm for training and testing the six CNN types to perform steganalysis task is given in below Algorithms 1 and 2.

Algorithm 1 **CNN Training**

```
Input: Cover, Stego train image dataset, with elements $\boldsymbol{I_i}$, $i$=1, 2, …, 2$N$ and labels $y_i$
Output: Trained Model
function CNN_Train
    Sample $M$ minibatches from { $I_i$, $\mathrm{y_i}$ }
    Initialize parameters
        iteration, $t$ ← 1
        weights ($w$) and bias ($b$) ← random gaussian distribution
        $update_1^W$ ← 0, $update_1^b$ ← 0
        learning rate $\eta$ ← 0.01
        momentum $\gamma$ ← 0.9
    while max(E) not reached do
        # forward pass
            pass $t^{th}$ minibatch to the network and compute predicted labels $\hat{y}_i$
            compute cross entropy loss ($\mathcal{L}$) #
                $$\mathcal{L}(w,b) = -\frac{1}{M}\sum_{i=1}^{n} y_i * \log(\hat{y}_i)^2$$
            objective function is to $minimize_{w,b}\ \mathcal{L}(w,b)$
        # backward pass
            compute gradients $\nabla w_t$ at $w_t$ and $\nabla b_t$ at $b_t$
                $$\nabla w_t = \frac{\partial(\mathcal{L}(w,b)}{\partial w}$$
                $$\nabla b_t = \frac{\partial(\mathcal{L}(w,b)}{\partial b}$$
            update weights and biases
                $$w_{t+1} = w_t - update_t$$
                $$update_t^w = \gamma * update_{t-1}^w + \eta * \nabla w_t$$
                $$b_{t+1} = b_t - update_t$$
                $$update_t^b = \gamma * update_{t-1}^b + \eta * \nabla b_t$$
            $t$ ← $t$ +1
    end while
    return model (updated $w$ and $b$)
end function
```

Algorithm 2 **CNN Testing**

```
Input: Trained Model, Test image dataset, with elements $\boldsymbol{I_j}$, j=1, 2, …, N
Output: Predicted class labels
function CNN_Test ($I_j$)
    model ← CNN_Train
    pred_label ← predict (model, $\boldsymbol{I_j}$)
    return pred_label
    end function
```

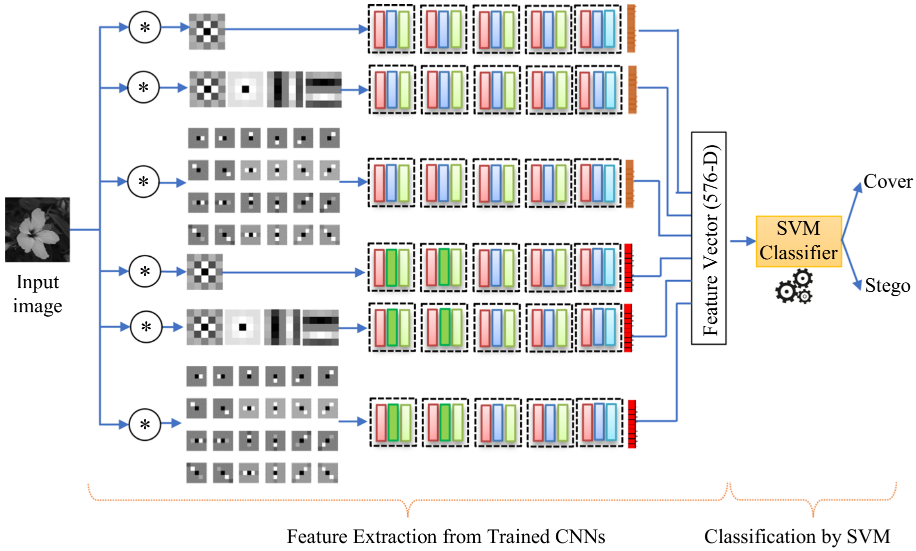


Fig. 2 Schematic of the proposed MixNet for generic steganalysis of content-adaptive steganography

3.2 Stage 2: MixNet Formation

Figure 2 shows the schematic of the proposed MixNet for generic steganalysis. After training the six CNN types, the trained networks are saved. Using the saved networks, features are extracted from the activations of global average pooling layer and are concatenated to form the final feature.

For the CNN Type I, II and III the dimension of the global average pool feature is 128 and for the rest of the models the dimension is 64. Overall, the feature dimension is 576 as mentioned in Fig. 2. Since there is a mixture of features taken from six CNN architectures, the network is named as 'MixNet'. The features are then classified by a Support Vector Machine classifier to provide the final classification result. The MixNet emphasizes the use of CNNs as a powerful feature extractor to learn and extract discriminating features from the noise residuals of stego and cover images.

4 Experiment Settings

The experiment settings to evaluate the performance of the proposed MixNet steganalysis framework is briefly presented. The database for cover images is selected from BOSSbase v1.01[1] [41] which consists of 10,000 raw, uncompressed grayscale images of resolution 512 × 512 pixels. Using Matlab 'imwrite' command, these raw images are converted to bitmap (BMP) format. The content-adaptive stego database is created by generating stego images by HUGO-BD, S-UNIWARD and WOW algorithms. The Matlab code for these algorithms is download from Binghamton University.[2] The payloads are random binary bits generated with five different relative payloads viz 0.1 bpp, 0.2 bpp, 0.3 bpp, 0.4 bpp and 0.5 bpp. Therefore,

[1] https://dde.binghamton.edu/download/.

[2] http://dde.binghamton.edu/download/stego_algorithms/.

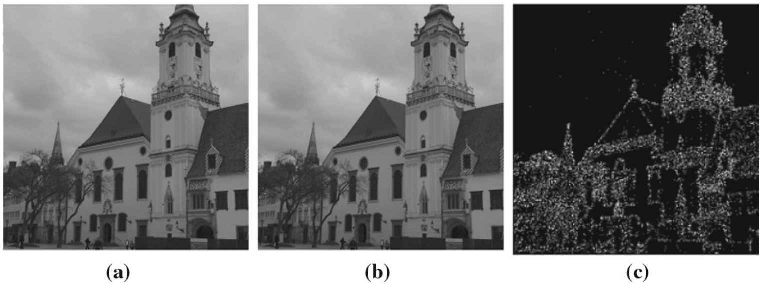


Fig. 3 Visualization of embedding pattern of HUGO-BD algorithm at 0.4 bpp. **a** Cover image. **b** Stego image. **c** difference image

for every relative payload bin, 10,000 stego images are created from the cover database using one algorithm resulting in a total 1,50,000 images in the stego database. A sample cover image, stego image (created by HUGO-BD at 0.4 bpp payload) and their difference image is shown in Fig. 3.

From Fig. 3 it can be clearly seen that, visually the stego image and cover image are looking alike but there are differences in the pixel values of stego image along the edges and some texture regions. This embedding pattern of HUGO-BD algorithm changes for every image based on its image contents. Similar pattern is observed for S-UNIWARD and WOW algorithms also. This highlights the difficulty in developing a discriminating feature from a group of cover and stego images. Hence in this work, CNNs are utilized to learn and then extract needful discriminating features.

Training the proposed six CNN models is done by using Stochastic Minibatch Gradient Descent backpropagation algorithm. The minibatch size is set as 64 with 32 cover images and their corresponding 32 stego images. The number of epochs used for training is 20 and a learning rate of 0.001 is fixed for all the epochs. The weights and bias in convolution and FC kernels are initialized to random zero-mean Gaussian distribution with standard deviation of 0.01 or leaky He initialization and later on updated while network training. In training phase, for every payload bin, six CNN models are trained using 8000 cover and 8000 stego images and tested with the remaining 2000 cover and 2000 stego images. The average elapsed time for training the CNN types is 50 min. The same set of training and testing images are then fed as input separately to the proposed MixNet to extract features and finally the extracted features of training images are used to train the SVM classifier. The performance of the proposed MixNet is then validated by detecting the features of testing images using the trained SVM classifier in testing phase.

5 Experimental Results and Discussion

The experiments to evaluate the proposed MixNet steganalysis framework are presented illustratively. First set of experiments are done to train the six CNN Types (1 to 6) so that those models learn to extract discriminating features of embedding distortions made by content-adaptive steganographic algorithms. The CNN models are executed in a workstation equipped with NVIDIA 32 GB GV 100 GPU card. The performance metrics used for evaluation is accuracy ((TP + TN)/(TP + FP + FN + TN)) and error rate (1 – accuracy) computed from confusion matrix (obtained from predicted and actual labels) where TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative. For steganalysis, stego class is positive and cover class is negative.

5.1 Evaluation of Proposed MixNet to Perform Generic Steganalysis

Table 3 presents the detection test accuracy (%) comparison of using the six types of CNN architectures individually and then as feature extractors in MixNet to detect the three content-adaptive steganographic algorithms.

Comparing the six CNN Types' results presented in Table 3, the CNN Type III achieves better accuracy in all the payload bins when compared to the other five types. HUGO-BD algorithm is detected best with accuracy of 81.625% and 59.55% in payload bins 0.5 bpp and 0.1 bpp respectively by CNN Type III. Similar kind of improvement is achieved for S-UNIWARD and WOW algorithms also.

SVM training in MixNet is carried out with tenfold cross validation and the average performance metrics is reported. The goal of performing cross-validation is to statistically train the classifier for generalization and prevent overfitting. The test accuracies tabulated in Table 3 also clearly shows the superior performance of the proposed MixNet when compared to the individual proposed CNN Types. In detecting HUGO-BD algorithm for 0.1 bpp, 0.2 bpp, 0.3 bpp, 0.4 bpp and 0.5 bpp payload bins, there is an increase in accuracy of 11.6%, 11.8%, 11.3%, 13.8% and 14.7% respectively achieved by MixNet when compared to the average accuracies of CNN Types. Similarly for S-UNIWARD detection, increase percentage observed are 12.7%, 12.7%, 14.6%, 12.8% and 14.7% and for WOW detection the increase percentage observed are 11.4%, 11.4%, 11.0%, 13.5% and 14.1% in the payload bins 0.1 bpp, 0.2 bpp, 0.3 bpp, 0.4 bpp and 0.5 bpp. Since in the proposed MixNet, features extracted

Table 3 Test accuracy (%) for detection of content-adaptive Steganographic algorithms by the proposed CNN types and MixNet

Algorithm	Payload (bpp)	Type I	Type II	Type III	Type IV	Type V	Type VI	Mix-Net
HUGO-BD	0.1	54.28	54.15	59.55	50.03	53.45	50.48	*65.25*
	0.2	64.43	63.73	66.40	53.68	63.20	60.83	*73.85*
	0.3	70.75	70.28	72.63	68.28	67.65	68.10	*80.95*
	0.4	75.65	71.58	76.90	74.03	70.78	73.53	*87.55*
	0.5	79.08	79.83	81.63	74.25	76.18	77.20	*92.73*
S-UNIWARD	0.1	50.00	50.63	56.03	50.00	50.03	50.58	*63.95*
	0.2	61.60	60.75	64.43	52.78	54.55	55.45	*71.00*
	0.3	68.48	67.18	70.98	56.33	66.43	65.13	*80.38*
	0.4	74.98	74.68	76.25	71.33	71.95	72.13	*86.33*
	0.5	79.58	79.20	80.75	77.18	77.58	79.50	*93.65*
WOW	0.1	50.08	53.80	57.10	50.03	50.78	50.40	*63.45*
	0.2	61.28	61.33	65.38	51.43	59.35	56.13	*70.55*
	0.3	66.73	68.25	70.35	64.48	65.05	64.80	*77.58*
	0.4	73.30	73.18	75.75	67.03	70.68	70.10	*85.20*
	0.5	78.55	77.43	79.20	76.83	76.88	75.13	*91.48*

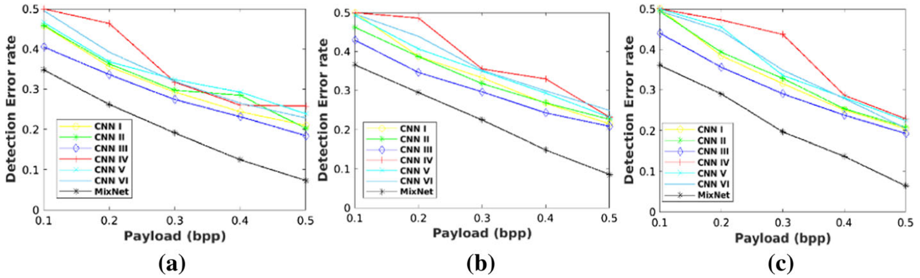


Fig. 4 Detection error rate comparison of all the proposed CNN types and MixNet to detect content-adaptive algorithms. a HUGO-BD. **b** S-UNIWARD. **c** WOW

by all the six CNN models are used to train the SVM classifier, there is a substantial increase in the detection accuracy. Figure 4 presents the detection error rate comparison graphs for the proposed methods.

From Fig. 4, it is evident that the content-adaptive steganographic algorithms are detected with minimal error rate using the proposed MixNet model which uses all the six CNNs to extract unique features for steganalysis from residues instead of images. Figure 5 presents the distribution of correctly identified cover and stego images while testing the MixNet framework across various payloads.

It is clearly observed from Fig. 5 that, on an average the number of correctly identified stego images is higher by almost 4.1%, 7.2%, 9.5%, 4.6% and 6.4% for the payload bins 0.1 bpp, 0.2 bpp, 0.3 bpp, 0.4 bpp and 0.5 bpp respectively.

In the proposed work, more importance is given to the feature extraction part, because, if data can be modelled in such a way that the extracted features are easily discriminable and less correlated, then any linear classifier will be able to classify the features. This will reduce the burden on the classifiers. In order to select an apt classifier for classifying the features extracted by the proposed MixNet, random forest, Adaboost and SVM classifiers

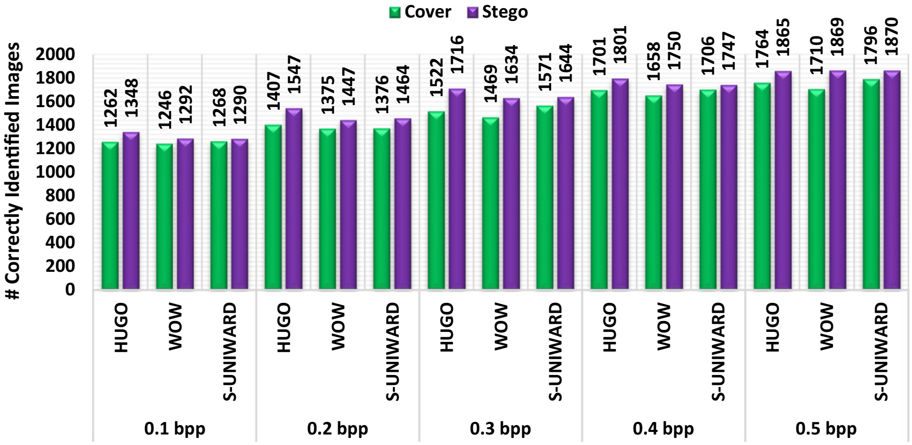
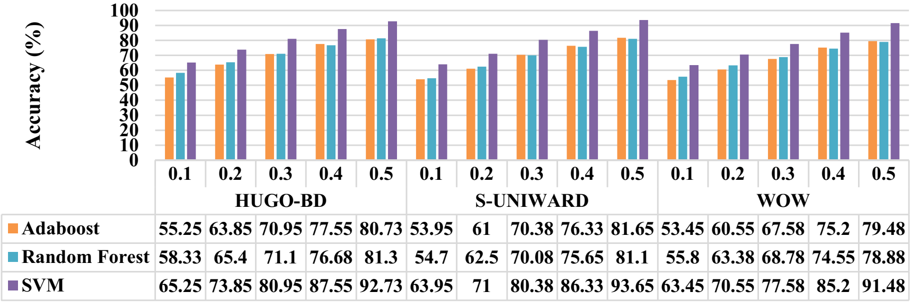


Fig. 5 Distribution of number of cover and stego images of content-adaptive steganographic algorithms that are correctly identified across various payloads by the Proposed MixNet steganalysis framework



	HUGO-BD					S-UNIWARD					WOW				
	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
🟧 Adaboost	55.25	63.85	70.95	77.55	80.73	53.95	61	70.38	76.33	81.65	53.45	60.55	67.58	75.2	79.48
🟦 Random Forest	58.33	65.4	71.1	76.68	81.3	54.7	62.5	70.08	75.65	81.1	55.8	63.38	68.78	74.55	78.88
🟪 SVM	65.25	73.85	80.95	87.55	92.73	63.95	71	80.38	86.33	93.65	63.45	70.55	77.58	85.2	91.48

Fig. 6 Comparison of using different classifiers in Stage 2 of MixNet framework

are experimented. Comparison of accuracy obtained by employing random forest, Adaboost and SVM classifiers is provided in Fig. 6.

Thus from Fig. 6, it is evident that the SVM classifier outperforms the Random forest and Adaboost classifiers. For example, in the low volume 0.1 bpp payload bin, the SVM classifier is able to achieve 65.25%, 63.95% and 63.45% for the detection of HUGO-BD, S-UNIWARD and WOW algorithms respectively whereas the random forest and Adaboost classifiers have obtained only 58.33%, 54.7%, 55.8% and 55.25%, 53.95% and 53.45% only. Therefore, the proposed low dimensional MixNet features along with SVM classifier can detect even low volume payloads with better accuracy. In the other slightly increased payloads also the SVM classifier gives better accuracy. Hence, the SVM classifier is incorporated in the proposed MixNet framework for accomplishing passive steganalysis.

5.2 Comparison of Proposed MixNet with State-of-the Art Feature Extractors

The performance of the proposed MixNet steganalysis framework results is compared with well-known feature extractors SRM [8] and maxSRMd2 [17] and a deep learning based steganalytic model Yu net [42] existing in the literature. In order to ensure a fair comparison, traditional and modern steganalyzers are both taken into account. Table 4 presents the detection error rates for passive steganalysis achieved by the proposed MixNet and the state-of-the-art steganalyzers.

From Table 4 it is inferred that, the detection error rate of the proposed MixNet steganalysis framework is low against HUGO-BD, S-UNIWARD and WOW algorithms when compared to both traditional feature extractors and deep learning model regardless of payload. Due to the robust mixture of features extracted by CNNs employed in the proposed MixNet, the proposed method outperforms the existing techniques.

The feature set dimension of SRM is 34,671 and maxSRMd2 is 12,753. The content-adaptive steganographic algorithms embed bits of secret payloads in hard-to-model regions of image like edges and textures in order to make unnoticeable changes to cover image while creating stego image. The proposed MixNet CNN models derive steganalytic features by capturing such delicate embedding distortions which are the underlying difference between any innocent cover image and illicit stego image. This is initially performed through the three different types of pre-processing filters that are employed to extract the residuals of images by suppressing the image content information. These residual images are then passed through sequential convolution layers that further capture the feeble differences between the cover and stego images. Compared to SRM and maxSRMd2, the proposed MixNet has a

Table 4 Detection error rate comparison with existing Steganalytic techniques

Algorithm	Method	0.1 bpp	0.2 bpp	0.3 bpp	0.4 bpp	0.5 bpp
HUGO-BD	SRM	0.364	0.2658	0.1955	0.1355	0.0854
	maxSRMd2	0.447	0.38	0.325	0.264	0.217
	Yu net	0.3711	0.2816	0.2194	0.1517	- NA-
	MixNet	*0.3475*	*0.2615*	*0.1905*	*0.1245*	*0.07275*
S-UNIWARD	SRM	0.4025	0.321	0.2495	0.2055	0.1664
	maxSRMd2	0.45	0.399	0.3195	0.26	0.204
	Yu net	0.4511	0.3319	0.2511	0.1845	- NA-
	MixNet	*0.3605*	*0.29*	*0.19625*	*0.13675*	*0.0635*
WOW	SRM	0.3977	0.3175	0.2492	0.2067	0.1623
	maxSRMd2	0.47	0.409	0.354	0.297	0.255
	Yu net	0.4577	0.2917	0.216	0.1847	- NA-
	MixNet	*0.3655*	*0.2945*	*0.22425*	*0.148*	*0.08525*

reduced feature dimension of 576-D. this reduction in feature dimension is achieved by the CNN type I to VI that has reduced layers when compared with existing CNN architectures in literature. Table 5 provides a comparison of space complexity (i.e., number of layers) of the proposed CNN model with state-of-the-art CNN steganalyzers. For better understanding, detection accuracy percentage for the detection of S-UNIWARD algorithm is also provided.

From Table 5 it is inferred that, compared to the Yu Net, Zhang Net and SRNet, the number of layers in CNN architecture of the proposed method is less. It is also inferred that with reduction in space complexity, the detection accuracy for the proposed MixNet is better than Yu Net and Zhang Net which has more number of layers. In the case of SRNet, the accuracy increase is approximately three percentage but it takes a vast increase in the number of layers to achieve this. But the proposed method could achieve a comparable performance to SRNet even with reduced layers. Thus, the proposed MixNet framework demonstrates a space efficient steganalyzer to achieve better detection of spatial content-adaptive steganographic algorithms.

Table 5 Comparison of number of layers present in the proposed CNN model and state-of-the-art CNN steganalyzers

Model	Number of layers	Accuracy (%) for the detection of S-UNIWARD 0.4 bpp
Yu net [42]	38	81.55
Zhang net [37]	62	74.61
SRNet [30]	100	89.77
MixNet	17	86.325

6 Conclusion

In this paper, a new steganalysis framework named MixNet with robust features that are extracted using six convolutional neural network architectures is demonstrated to be a credible tool for steganalysis of spatial content-adaptive algorithms. The developed MixNet achieves 92.73%, 93.65% and 91.48% accuracy in detecting HUGO-BD, S-UNIWARD and WOW respectively in 0.5 bpp payload bin. On the whole, the proposed MixNet with less feature dimension and reduced space complex CNN structure is significantly better in detecting content-adaptive algorithms.

Since the proposed MixNet framework is designed with convolutional neural networks, the input image size is fixed to 512 × 512. So, the framework will not accept images other than this prescribed size and in such case, images must be resized. This resizing may lead to loss of information. In future work, the MixNet will be adapted to accept images of arbitrary sizes in order to detect any test image of arbitrary size. Also, the proposed MixNet will be extended to detect transform domain content-adaptive algorithms.

Acknowledgements The authors would like to thank ER & IPR-DRDO New Delhi, Management and Principal of Mepco Schlenk Engineering College, Sivakasi for providing the necessary facilities and support to carry out this research work. The authors would also like to thank anonymous reviewers for their insightful comments that led to significant improvements of the manuscript.

Funding The work presented in this paper was funded by Directorate of Extramural Research & Intellectual Property Rights (ER & IPR), Defence Research Development Organization (DRDO), Ministry of Defence, Government of India under Grant No. ERIP/ER/201702007/M/01/1733. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of Indian Government.

References

1. Fridrich J (2010) Steganography in digital media: principles, algorithms, and applications. Cambridge University Press, Cambridge
2. Zielińska E, Mazurczyk W, Szczypiorski K (2014) Trends in steganography. Commun ACM 57:86–95. https://doi.org/10.1145/2566590.2566610
3. Ker AD, Bas P, Böhme R, et al (2013) Moving steganography and steganalysis from the laboratory into the real world. In: Proceedings of the first ACM workshop on Information hiding and multimedia security-IH&MMSec'13. ACM Press, Montpellier, p 45
4. Sedighi V, Cogranne R, Fridrich J (2016) Content-adaptive steganography by minimizing statistical detectability. IEEE TransInformForensic Secur 11:221–234. https://doi.org/10.1109/TIFS.2015.2486744
5. Nissar A, Mir AH (2010) Classification of steganalysis techniques: a study. Digital Signal Process 20:1758–1770. https://doi.org/10.1016/j.dsp.2010.02.003
6. Babu J, Rangu S, Manogna P (2017) A survey on different feature extraction and classification techniques used in image steganalysis. JIS 08:186–202. https://doi.org/10.4236/jis.2017.83013
7. Denemark T, Boroumand M, Fridrich J (2016) Steganalysis features for content-adaptive JPEG steganography. IEEE TransInformForensic Secur 11:1736–1746. https://doi.org/10.1109/TIFS.2016.2555281
8. Fridrich J, Kodovsky J (2012) Rich models for steganalysis of digital images. IEEE TransInformForensic Secur 7:868–882. https://doi.org/10.1109/TIFS.2012.2190402
9. Pevný T, Filler T, Bas P (2010) Using high-dimensional image models to perform highly undetectable steganography. In: Böhme R, Fong PWL, Safavi-Naini R (eds) Information hiding. Springer, Berlin, pp 161–177
10. Arivazhagan S, Sylvia Lilly Jebarani W, Veena ST, Shanmugaraj M (2015) A novel low-D feature based generic steganalyzer to detect low volume payloads. Indian J Sci Technol. https://doi.org/10.17485/ijst/2015/v8i24/79991

11. Tang W, Li H, Luo W, Huang J (2015) Adaptive steganalysis based on embedding probabilities of pixels. IEEE TransInformForensic Secur. https://doi.org/10.1109/TIFS.2015.2507159
12. Holub V, Fridrich J (2012) Designing steganographic distortion using directional filters. In: 2012 IEEE international workshop on information forensics and security (WIFS). IEEE, Costa Adeje-Tenerife, pp 234–239
13. Filler T, Fridrich J (2010) Gibbs construction in steganography. IEEE TransInformForensic Secur 5:705–720. https://doi.org/10.1109/TIFS.2010.2077629
14. Holub V, Fridrich J (2013) Digital image steganography using universal distortion. In: Proceedings of the 1st ACM workshop on information hiding and multimedia security-IH&MMSec'13. ACM Press, Montpellier, p 59
15. Yu J, Zhang X, Li F (2016) Spatial steganalysis using redistributed residuals and diverse ensemble classifier. Multimed Tools Appl 75:13613–13625. https://doi.org/10.1007/s11042-015-2742-y
16. Boroumand M, Fridrich J (2018) Applications of explicit non-linear feature maps in steganalysis. IEEE TransInformForensic Secur 13:823–833. https://doi.org/10.1109/TIFS.2017.2766580
17. Denemark T, Sedighi V, Holub V et al (2014) Selection-channel-aware rich model for Steganalysis of digital images. In: 2014 IEEE international workshop on information forensics and security (WIFS). IEEE, Atlanta, pp 48–53
18. Li B, Li Z, Zhou S et al (2018) New steganalytic features for spatial image steganography based on derivative filters and threshold LBP operator. IEEE TransInformForensic Secur 13:1242–1257. https://doi.org/10.1109/TIFS.2017.2780805
19. Lu J, Zhou G, Yang C et al (2019) Steganalysis of content-adaptive steganography based on massive datasets pre-classification and feature selection. IEEE Access 7:21702–21711. https://doi.org/10.1109/ACCESS.2019.2896781
20. Wang P, Liu F, Yang C (2020) Towards feature representation for steganalysis of spatial steganography. Signal Process 169:107422. https://doi.org/10.1016/j.sigpro.2019.107422
21. Tan S, Li B (2014) Stacked convolutional auto-encoders for steganalysis of digital images. In: Signal and information processing association annual summit and conference (APSIPA), 2014 Asia-Pacific. IEEE, Chiang Mai, pp 1–4
22. Pibre L, Pasquet J, Ienco D, Chaumont M (2016) Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover sourcemismatch. Electron Imaging 2016:1–11. https://doi.org/10.2352/ISSN.2470-1173.2016.8.MWSF-078
23. Xu G, Wu H-Z, Shi Y-Q (2016) Structural design of convolutional neural networks for steganalysis. IEEE Signal Process Lett 23:708–712. https://doi.org/10.1109/LSP.2016.2548421
24. Qian Y, Dong J, Wang W, Tan T (2018) Feature learning for steganalysis using convolutional neural networks. Multimed Tools Appl 77:19633–19657. https://doi.org/10.1007/s11042-017-5326-1
25. Couchot J-F, Guyeux C, Couturier R, Salomon M Steganalysis via a convolutional neural network using large convolution filters for embedding process with same stego key. 25
26. Zhang R, Zhu F, Liu J, Liu G (2018) Efficient feature learning and multi-size image steganalysis based on CNN. arXiv:180711428 [cs]
27. Wu S, Zhong S, Liu Y (2018) Deep residual learning for image steganalysis. Multimed Tools Appl 77:10437–10453. https://doi.org/10.1007/s11042-017-4440-4
28. Wu S, Zhong S, Liu Y (2017) Residual convolution network based steganalysis with adaptive content suppression. In: 2017 IEEE international conference on multimedia and expo (ICME). IEEE, Hong Kong, pp 241–246
29. Wang W, Dong J, Qian Y, Tan T Deep steganalysis: end-to-end learning with supervisory information beyond class labels. 11
30. Boroumand M, Chen M, Fridrich J (2019) Deep residual network for steganalysis of digital images. IEEE TransInformForensic Secur 14:1181–1193. https://doi.org/10.1109/TIFS.2018.2871749
31. Zhang S, Zhang H, Zhao X, Yu H (2019) A deep residual multi-scale convolutional network for spatial steganalysis. In: Yoo CD, Shi Y-Q, Kim HJ et al (eds) Digital forensics and watermarking. Springer, Cham, pp 40–52
32. Kim J, Park H, Park J-I (2020) CNN-based image steganalysis using additional data embedding. Multimed Tools Appl 79:1355–1372. https://doi.org/10.1007/s11042-019-08251-3
33. Ren W, Zhai L, Jia J et al (2020) Learning selection channels for image steganalysis in spatial domain. Neurocomputing 401:78–90. https://doi.org/10.1016/j.neucom.2020.02.105
34. Ye J, Ni J, Yi Y (2017) Deep learning hierarchical representations for image steganalysis. IEEE TransInformForensic Secur 12:2545–2557. https://doi.org/10.1109/TIFS.2017.2710946
35. Tan S, Wu W, Shao Z et al (2020) CALPA-NET: channel-pruning-assisted deep residual network for steganalysis of digital images. arXiv:191104657 [cs, eess]

36. Luo W, Li J, Yang J et al (2017) Convolutional sparse autoencoders for image classification. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2017.2712793
37. 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 Inform Forensic Secur 15:1138–1150. https://doi.org/10.1109/TIFS.2019.2936913
38. Liu F, Zhou X, Yan X et al (2021) Image steganalysis via diverse filters and squeeze-and-excitation convolutional neural network. Mathematics 9:189. https://doi.org/10.3390/math9020189
39. Ruan F, Zhang X, Zhu D et al (2020) Deep learning for real-time image steganalysis: a survey. J Real-Time Image Proc 17:149–160. https://doi.org/10.1007/s11554-019-00915-5
40. Selvaraj A, Ezhilarasan A, Wellington SLJ, Sam AR (2021) Digital image steganalysis: A survey on paradigm shift from machine learning to deep learning based techniques. IET Image Process 15:504–522. https://doi.org/10.1049/ipr2.12043
41. Bas P, Filler T, Pevný T (2011) ”Break our steganographic system”: the ins and outs of organizing BOSS. In: Filler T, Pevný T, Craver S, Ker A (eds) Information hiding. Springer, Berlin, pp 59–70
42. Yu X, Tan H, Liang H et al (2018) A multi-task learning CNN for image steganalysis. In: IEEE international workshop on information forensics and security (WIFS), pp 1–7. https://doi.org/10.1109/WIFS.2018.8630766.

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

Terms and Conditions