Journal of Information Hiding and Multimedia Signal Processing ©2020 ISSN 2073-4212
Ubiquitous International Volume **11**, Number **3**, September **2020**

Enhancing CNN-based Image Steganalysis on GPUs

Eslam M. Mustafa, Mohamed A. Elshafey and Mohamed M. Fouad

Department of Computer Engineering
Military Technical College, Kobry ELkobba, Cairo, Egypt
eng.eslam.mtc@gmail.com, m.shafey@mtc.edu.eg, mmafoad@mtc.edu.eg

Received May 2020; Accepted July 2020

Abstract. *Blind image steganalysis (BIS) is the process of detecting whether an input image has hidden data or not, without any prior known information (*i.e.*, blind) on the applied steganography technique. Recent BIS approaches typically suffer from limited detection accuracy and higher computational cost due to,* e.g.*, pre-processing. In this paper, the proposed BIS approach discards the pre-processing step, so that the computational cost is reduced. As well, significant modifications on a recent convolution neural network (CNN)-model are considered in order to enhance the detection accuracy. First, an efficient parameters initialization is considered. Second, a cyclic learning rate and the LReLU activation function are used, during the learning phase, for faster convergence with noticeably higher detection accuracy. Finally, a hybrid technique of model and data parallelism techniques is performed in both convolution and fully connected layers, respectively, thus significantly reducing the computational cost. Given stego-images exposed to the S-UNIWARD, WOW and HILL steganography techniques with different payloads, the results show that the proposed approach outperforms competing approaches by an enhancement of up to 13.3% and 21.2X in terms of detection accuracy and speed-up factor, respectively.*
Keywords: Image steganalysis, Deep learning, Data & model parallelism, GPUs.

1. **Introduction.** Despite the remarkable success achieved in steganography techniques, there is an urgent need to eliminate its negative use of by the process of steganalysis. This latter detects the existence of any hidden data (*i.e.,* secret data) or even more to extract these data. The multimedia cover in which steganographers can hide data may be text, image, audio or video [1]. This paper is concerned with images as their cover due to their prevalence and widespread use on the Internet, especially on social media.

Blind image steganalysis attempts to distinguish the presence of a shrouded message in a digital image without any prior information on the steganographic technique used [2]. The main challenge in image steganalysis is to extract the stego-noise [3]. CNNs can be used in the image steganalysis issue due to its adaptive ability to detect the modern steganography techniques. Generally, traditional CNN-based image steganalysis approaches yield reasonable detection accuracy for high-payloaded stego-images, however, the detection accuracy is negatively impacted in case of detecting stego-images with low payload. In such cases, the CNN model used requires two levels of training, thus increasing the training time needed. Moreover, other factors would affect the detection accuracy, such as the pre-processing, feature extraction and classification phases. Typically, a high pass filter (HPF) with fixed coefficients is used as a pre-processing phase. However, that filtering would increase the computational cost, the fixed filter coefficients are not necessarily to match all input stego-images, thus decreasing the detection accuracy. In addition, the current BIS approaches use their CNN model with non-initialized parameters, fixed learning rate, and Gaussian activation function during the learning phase, thus negatively affecting the detection accuracy. Nevertheless, most of the modern BIS approaches use either data parallelism or model parallelism during the learning phase, thereby the advantages of one of them is discarded. Generally speaking, the recent CNN-based BIS approaches present a limited enhancement in the detection accuracy with a noticeable increase in the training time.

This paper goal is to produce a highly accurate blind image steganalysis technique based on the well-known image steganalysis technique IGNCNN [4] with set of major modifications in order to efficiently detect stego-images that may contain harmful hidden data. In this paper, the proposed BIS approach omits the pre-processing step, thus decreasing the computational cost and initialize the first convolutional layer with a set of high pass filters that help enhancing the stego-noise extraction process. In addition, significant modifications, on the CNN-model of the most recent BIS approaches, are carried out in order to enhance the detection accuracy. First modification is to use an efficient parameters initialization. Second, during the learning phase, a cyclic learning rate and the LReLU activation function are used for faster convergence with noticeably higher detection accuracy. Finally, a hybrid parallelism technique is performed using both model and data parallelism techniques in both convolution and the fully connected layers, respectively, thereby the detection accuracy is significantly enhanced and the training time is reduced.

The rest of this paper is organized as follows. The related work to the CNN-based image steganalysis approaches and a focus on the most recent one, namely QianNet [5], are presented in Sec. 2 and Sec. 3, respectively. Then, the proposed approach and its implementation on two graphical processing units (GPUs) are explained Sec. 4 and Sec. 5, respectively. Experiments and results are shown in Sec. 6. Finally, conclusions are drawn in Sec. 7.

2. **The Related Work and Problem Statement.** Blind image Steganalysis is defined as the process of detecting whether an image contains secret information (binary classification problem). This problem can be divided into two main steps: i) extracting features and ii) classifier training. Traditional image steganalysis approaches, such as Spatial Rich Models (SRM) [6, 7] perform the feature extraction step using handcrafted feature extractors. These extractors are very difficult to be designed, particularly with the rapid increase in the complexity of steganographic techniques. Traditional steganalysis approaches use classifiers, such as Fisher Linear Discriminant (FLD)-based ensemble classifier and the support vector machine (SVM) [7] to perform the second step. The separation between the two steps in these approaches makes the simultaneous optimization for the two steps impossible.

CNN-based steganalysis approaches are proposed to avoid the disadvantages of traditional steganalysis ones [8]. These approaches can automatically extract features from input images. Classification and feature extraction are unified under a single model which makes it possible to optimize both classification and feature extraction steps simultaneously. CNN-based steganalysis approaches present comparable results with the traditional approaches. In [8], a blind CNN-based image steganalysis approach is firstly presented. Although that approach's detection accuracy reaches comparable results to that of [6], the former outperforms the results in SPAM [9]. The CNN in [8] consists of nine convolutional layers and one fully connected layer. Max Pooling is used as a pooling layer and Softmax is used for classification module. No pre-processing layer is implemented. No batch normalization nor absolute layer is used.

Gaussian Neuron CNN (GNCNN) [10] is the first CNN-based approach with results comparable to SRM results. In order to strengthen the noise signal, a HPF is used. That CNN consists of five convolutional layers and three fully connected layers. It also uses a Gaussian function as an activation layer. GNCNN [10] consists of one pre-processing layer, five convolutional layers and three fully connected layers. Average Pooling is used as pooling layer, a Gaussian activation function is used in the hidden layers and Softmax is used for classification module. No batch normalization or absolute layer are used.

Based on the GNCNN [10], authors in [11] present a CNN-based image steganalysis approach but with different structure. This network has only two higher convolutional layers. The Gaussian activation layer is also replaced by ReLU function [12]. The pooling layers are not used. The training of this network is performed under two scenarios Clairvoyant and Cover-Source Mismatch [11]. The results of this network outperform the results of SRM.

Authors in [13] present a CNN-based steganalysis approach called XuNet. XuNet [13] consists of one pre-processing layer, five convolutional layers and two fully connected layers. Average pooling is used as a pooling layer, Tanh [14] and ReLU [12] activation functions are used in the hidden layers, and Softmax is used for classification module. Batch normalization is used, and an absolute layer exists after the first convolutional layer. The approach in [13] enhances the statistical modeling by using an ABS layer and 1×1 convolution kernels. This approach also uses batch normalization layer in order to prevent the network stuck in a poor minima and enhance the updating process of the biases parameters. That approach uses two activation functions, the Tanh [14] function in the first two layers and the ReLU [12] in the remaining layers, in order to avoid the over-fitting problem. In [15], authors present an CNN-based approach to enhance the accuracy and precision of the classification operation and recoup the lost information. That improvement has been achieved due to using pooling operation by training a group of CNNs with a similar structure to that in XuNet, but with little modifications. These modifications are to increase the kernel size of the last two pooling layers and increase the number of convolutional layers by one.

In [4] and our earlier work [16, 17, 18], transfer learning is used to enhance the performance of GNCNN. Their results show that transferring pre-trained CNN features for detecting high payload stego-images of one steganographic algorithm can improve feature learning for detecting lower payload stego-images of that specific steganographic algorithm; named Improved Gaussian Neuron CNN (IGNCNN). In [19], authors improve CNN-based steganalysis detection accuracy by using the trained CNN to calculate image statistical model derivatives with respect to changes presented due to payload embedding.

To improve the detection accuracy of stego-images with low payload, [20] concludes that the larger the filter size exposed to the final convolutional layer of a CNN, as well as the less the filter size exposed to the remaining layers, the detection accuracy of stego-images, with large image size and lower payloads, will be accordingly enhanced. In [21], two CNNs are presented; the first is called generator, whereas the second is named discriminator. The generator CNN is a steganography CNN that hides information in specific locations that steganalyzers can not easily detect. While the discriminator CNN is trained on those locations to enhance the detection accuracy metric.

YeNet CNN-based steganalysis approach is introduced in [22]. YeNet uses a set of trainable HPFs instead of using the traditional ones for noise extraction process. Those trainable filters are initialized with the coefficients of SRM filters. The TLU activation layer is presented for first time in steganalysis to increase the signal-to-noise ratio (SNR). Yedroudj-Net, an image steganalysis approach presented in [23], is a combination between XuNet and YeNet. It uses the TLU activation function [22] and batch normalization

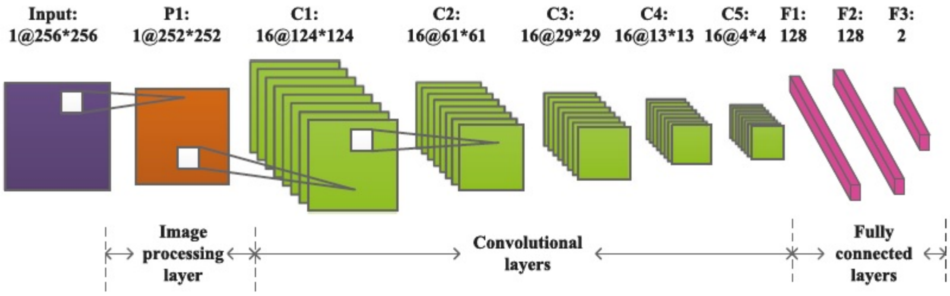

FIGURE 1. QianNet CNN structure [5].

layer. It also uses the SRM filter for initializing the values of the first convolutional layer. Simply it uses all the best configurations in XuNet and YeNet. YedroudjNet [23] network structure is presented as: 1 convolutional layer with 30 filters as a pre-processing layer, 5 convolutional layers as feature extractors and 3 fully connected layers. Average Pooling is used after all convolutional layers except for the first one. It uses two activation functions in the hidden layers which are TLU [22] and ReLU [12], and Softmax for classification. It also uses batch normalization and an absolute layer exists after the first convolutional layer. Authors in [24] update the YeNet to effectively detect steganography in images with high- resolution. The CNN is trained on small resolution images in order to adopt the network to high resolution ones. Another CNN model for steganalysis called ZhuNet is presented in [25]. ZhuNet optimizes the parameters of the pre-processing layer for the first time.The size of convolution kernel is reduced. In order to capture more relative features separable convolutions [26] are used. Spatial pyramid pooling (SPP) [27] is used to enhance the rendering of features and adopt the network to small images. In [5], QianNet is presented as a protraction of the work in [10] and [4]. This approach aims to enhance the extraction of the stego-noise which is a must in order to improve the detection accuracy. This enhancement is achieved by investigating the effect of using 3 different HPF from the set of filters used in SRM [6] (see 1). In order to boost the detection accuracy of the QianNet [5] model combination is used. Despite the enhancement achieved by this approach as well as aforementioned ones, the QianNet still has some challenges to enhance the detection accuracy with an enhanced training time, such as

- Using a HPF with fixed coefficients, especially for low payload stego-images.
- Using fixed learning rate, which may cause training process stuck in local minima.
- Using both Gaussian activation function that yields increasing the training time, and ReLU activation function that neglects the negative part of the feature maps.
- Using single GPU implementation that imposes either CNN data parallelism or CNN model parallelism, thus discarding the benefit of either of them.

3. **QianNet CNN-Structure: An Overview.** In this section, we focus on showing the structure of the QianNet [5] model. Basically, it includes a pre-processing layer to extract the stego-noise the output of this layer is fed as an input to the feature extraction module. This module consists of five convolutional layers, each of them contains 16 neurons. The extracted features are used as an input to the classification module which is consisted of two fully connected layers each has 128 neurons and one softmax layer for the classification decision. QianNet [5] CNN structure is shown in 1.

3.1. **The pre-processing Layer.** This layer is a HPF that extracts stego-noise from the stego-image. QianNet [5] train different CNNs having the same structure, however, with different HPFs as a pre-processing layer. These filters are four different filters elected out of the SRM [6] feature extractor filters. Those filters are $K_{5\times5}$, $K_{3\times2}$, $K_{1\times4}$ and $K^{max}_{5\times5}$.

3.2. **Model Combination.** Once the training of each CNN has completed, the CNN model with the lowest detection error is elected. Then, that model is added to other models that use different HPFs to greedily minimize the error generated per iteration.

3.3. **Activation Function.** QianNet [5] investigates four types of activation functions which are Gaussian, 1-Gaussian, ReLU, and Tanh. The lowest detection error is achieved when using Gaussian and ReLU activation functions respectively. However, the Gaussian activation function is the most complex activation function, thus leads to a noticeable increase in the training time. On the other hand, in despite of that the ReLU activation function is simpler and saves the training time, it has a negative effect by losing some data due to neglecting the negative parts of the feature maps.

4. **The Proposed CNN-based Steganalysis Approach.** In this section, the proposed approach is explained. This approach tackles the challenges of detection accuracy and training time of QianNet [5] in order to produce a highly accurate CNN-based image steganalysis approach with an accelerated training time.

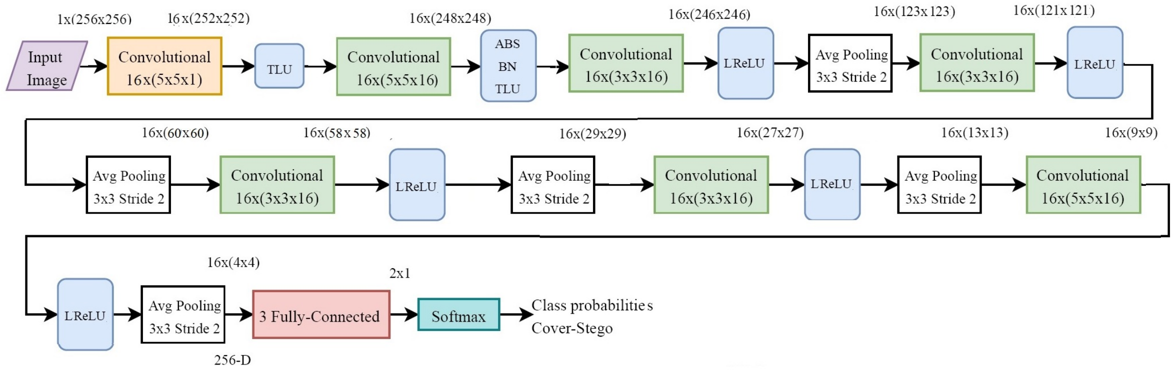

FIGURE 2. The proposed CNN model.

4.1. **The Architecture of the Proposed CNN Model.** The proposed model consists of two main modules: features extraction module and classification module. A pre-processing layer, for extracting stego-noise residuals, is embedded at the beginning of the features extraction module.

- Features Extraction Module consists of seven convolutional layers C0 - C6. Each of these layers consists of 16 neuron. The first layer C0 acts as the required pre-processing layer for extracting stego-noise residuals. The convolutional kernel in the C0, C1 and C6 is of size 5×5 and of size 3×3 in the rest. Avg pooling layer of window size 3×3 and stride 2 is applied after all convolutional layers except for the first two layers ,*i.e.,* , C0 and C1. All convolutional layers are followed by a LReLU activation function except for the first two layers, C0 and C1, which are followed by TLU activation function. As in XuNet [13] and YedroudjNet [23], an Absolute Layer (ABS) is placed after the first feature extraction layer (C1). ABS layer imposes the generated features to consider the sign symmetry, which may exist in the stego-noise. This layer is very effective in preventing the network from being stuck in a poor local minima and learning optimal biases. The Batch Normalization layer (BN) is used after the ABS layer as in XuNet [13], due to its ability to normalize the feature maps to a unit variance and a zero mean. This layer is also used to help the CNN to increase its insensitivity to parameter initialization and to faster convergence, which can be happened by compelling the input data from being in saturation regions [28]. In the proposed model the input image is converted into 256D feature vector by convolutional and pooling layers. The proposed network structure is shown in Fig.2.

- Classification module: This module consists of three fully connected layers: F1, F2 and F3. Layers F1 and F2 contain 128 ReLU [12] activated neurons, and F3 have 2 neurons with a two-way Softmax for classification.

4.2. **Efficient pre-processing Initialization.** In the proposed approach, instead of using a pre-processing layer, a special convolutional layer C0 is used. The parameters of this layer are not randomly initialized, instead, they are initialized with the basic high-pass filters used in computation of residual maps in SRM [6]. By using this special initialized parameters, the layer extracts the stego-noise residuals as the SRM [6], just for the first image. Then, these parameters are fine tuned every back-propagation pass, i.e., these parameters are adopted depending on the input images in order to effectively extract the stego-noise. This way for extracting the stego-noise is dataset dependant as the weights are updated based on the calculated error from directly the input images.

SRM filters are 30 filters divided into 6 classes: 1st, 2nd, 3rd, SQUARE, EDGE 3×3 and EDGE 5×5. These filters are distributed as follows: 8 filters in class 1st, 4 filters in class 2nd, 8 filters in class 3rd, 4 filters in class EDGE 3×3, 4 filters in class EDGE 5×5 and 1 filter in class SQUARE 3×3 and another 1 filter in class SQUARE 5×5. The maximum kernel size is 5×5.

4.3. **Implemented Activation Functions.** In the proposed CNN-based model, two activation functions, LReLU and TLU, are implemented [29]. LReLU activation function is a modified version of ReLU activation function, which effectively can select useful signal from input images i.e. extract sparse feature representations. These spares representations are linearly separable and have better generalization ability [29]. However, LReLU activation function has a small slope in the negative area, so can avoid negative data loss, which can be occured with implementation of ReLU. Using TLU activation function in the first convolution layers allows to perfectly deal with the dispersion of stego-noise, and hence the CNN can be learned effectively with HPFs in the first layer [29].

LReLU and TLU are characterized by the lowest complexity when compared to other nonlinear activation functions, *e.g.,* the Gaussian activation function. So, applying both LReLU and TLU activation functions instead of the Gaussian one, in the proposed approach, is suggested to accelerate the training process.

4.4. **CNN Learning Rate.** The learning rate is considered as one of the important hyper-parameters to be tuned for training deep neural networks. A stochastic gradient descent optimizer is typically used to update the weights in the training process. Weights are updated as, $\theta^t = \theta^{t-1} - \epsilon_t \frac{\partial L}{\partial \theta}$, where θ denotes weights, L denotes a loss function and ϵ_t denotes learning rate. Low learning rate justifies a reliable training, but on the other side, the optimization process consumes much time to reach minimum of the loss function. In case of a high learning rate, the training may neither converge nor diverge. So, the optimizer may miss the minimum and make the loss worse.

In the proposed approach, a method [30] for detecting the range of learning rates, is implemented. In the implemented method, the learning rate range, min_{LR} through max_{LR}, is selected by such a way , in which the loss decreases fast. In the defined learning rate interval, high learning rate may lead the system to traverse the saddle points faster and hence negatively affect the minimizing process [31]. On the other side, applying tiny gradients of these points decelerates the learning process. So, in the proposed CNN-based model the cyclical learning rate method is applied. In that cyclical version, the learning rate is started with a relatively high value, then it will be decreased gradually to tiny values when approaches the required optimal value. By applying cyclic learning rate method, a near optimal learning rate can be selected and implemented during the training process of the proposed approach [30].

5. **Implementing The Proposed CNN-Based Steganalysis Approach with Model Parallelism & Data Parallelism on GPUs.** In the proposed CNN-based approach, both model parallelism and data parallelism are applied in training process on GPUs in accordance with a variable batch size.

5.1. **Model Parallelism and Data Parallelism.** CNN-based image steganalysis approaches are models, which consist of multiple layers of multiple neurons, i,e, big model. These models are trained on large number of images, i.e, large dataset. Based on that, two parallelization methods can be used for parallelizing training process of such CNN models: i) model parallelism and ii) data parallelism.

- In model parallelism, the CNN model can be divided into different parts. Each part is trained on different GPU, that accelerates the training process. Generally, Model parallelism is effectively used when tasks assigned to a neuron are very large.
- In data parallelism, the full model is assigned to each of the used GPUs to allow each GPU to train the model on a different set of training data. Like model parallelism, data parallelism accelerates the training process. Data parallelism is effectively used when the computation process per network parameter is heavy.

In model parallelism, the workflows of the process on multiple GPUs, are dependent, in which each GPU requires output from the others. Accordingly, all the GPUs should be synchronized. In contrast, in data parallelism, the model parameters must be synchronized for all GPUs to train a consistent model.

The proposed CNN contains two types of layers: convolutional and fully connected layers. Most of computations (about 90-95%) are performed in convolutional layers, while fully-connected ones are those layers in which the neuron activity (parameters) is high (about 95%) [32]. Based on that, in the proposed CNN-based steganalysis approach, data parallelism can be effectively applied in the convolutional layers and model parallelism is more effective in the fully connected layers.

5.2. **Variable Batch Size.** Batch size affects the efficiency of data parallelism, in which, CNN parameters have to be synchronized to verify model consistency. Applying large batch size decreases the number of times required for synchronizing parameters (synchronization once per batch). On the other side, large batch size may negatively affect the detection accuracy and the convergence rate of the Stochastic Gradient Descent (SGD) of the training process. In the proposed CNN-based model, a batch of size 128 is applied in the fully-connected layers and of size $128 \times G$ for convolutional layers, where G is number of used GPUs. This variable batch size provides model capability to update the parameters of the fully-connected layers after each partial backward pass, with no extra computational cost. Additionally, The variable batch size, which is relatively large in convolutional layers, increases the efficiency of data parallelism.

5.3. **Training Workflow for The Proposed CNN-Based Steganalysis Approach.** The proposed CNN-based image steganalysis approach is trained in parallelized form using two GPUs, in which model parallelism and data parallelism are applied to the fully connected layers and convolutional layers, respectively. A batch of size 128 is applied in the

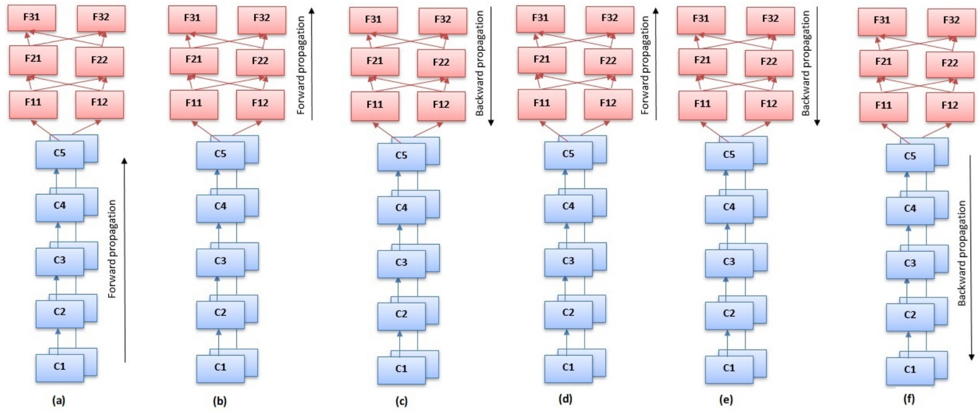

Figure 3. Illustration of the forward and backward propagation of the proposed approach using two GPUs. The standard two passes are replaced here with six passes (a, b, c, d, e and f).

fully connected layers and of size 256 in the convolutional layers. First, two batches of size 128 are passed to each GPU in order to perform feature extraction in the convolutional layers. Then, the first GPU passes its features map to the second GPU. Both GPUs perform the classification step based on the model parallelism. After error calculating for the first batch, partial backpropagation is performed in order to update the parameters of the fully-connected layers. At the end of partial backpropagation in the fully-connected layers, the second GPU passes its features map to the first and the whole process is carried out repeatedly. Finally, the backpropagation is performed to the convolutional layers and the CNN model is then ready to be trained on the next upcoming batches as shown Fig3.

6. Experimental Work and Results.

6.1. Implementation Setup. Experimental tests are performed on a device with two Intel Xeon Silver, 128 GB RAM and two Tesla V100 GPUs each with 5120 CUDA cores. A batch of size 256 and 128 for convolutional and fully-connected layers respectively are applied with a momentum of value 0.9. Weight decay is 0 for convolutional layers and 0.01 for the fully connected layers. Maximum learning rate Max_{LR}, in the effective learning rate range, equals 0.01 for training the CNN on S-UNIWARD stego-images and 0.001 for training the CNN on WOW and HILL stego-images, while minimum learning rate Min_{LR} equals 0.00001. Synchronization process between the GPUs is achieved using the built in SYNC-FUNCTION implemented in CUDA-convnet2.

In the experiments, the detection error (P_E) is used as the performance evaluation metric on the basis of the lower, the better. P_E is computed as shown in 1, where P_{FA} denotes the false alarm rate, determined as shown in Eq. 2. As well, FP and TN denote the false positive and the true negative, respectively. While P_{MD} denotes the missed detection rate that can be formulated as shown in Eq. 3. Also, P_D denotes the detection rate and can be cast as shown in Eq. 4. Finally, TP and FN denote the true positive and the false negative, respectively.

$$P_E = min_{P_{FA}} 0.5(P_{FA} + P_{MD}(P_{FA})) \tag{1}$$

$$P_{FA} = FP/(FP + TN) \tag{2}$$

$$P_{MD} = 1 - P_D \tag{3}$$

$$P_D = TP/(TP + FN) \tag{4}$$

The experimental results of the proposed CNN-based steganalysis approach will be compared with those of a set of eight recent competing steganalysis approaches: SRM with the Ensample classifier (SRM+EC) [7], GNCNN [10], IGNCNN [4], XuNet [13], YeNet [22], QianNet [5], YedroudjNet [23] and ZhuNet [25].

6.2. **Dataset Description.** In order to evaluate the performance of the proposed approach, two datasets are used. The first is the standard BOSSbase 1.01 image dataset [33] and is used for verification. BOSSbase 1.01 contains 10000 gray-level cover images of size 512×512, those images are acquired by seven cameras. All the BOSSbase 1.01 images are resized to 256×256 pixels using Matlab. Whereas, the second dataset is ImageNet [34], a data set of over 15 million high-resolution images collected from the web. 100,000 images are selected randomly, then converted them to 8-bit grayscale, and finally cropped out the central 256×256 from the resulting images. The second image dataset is used for validation without training (*i.e.,* testing only).

For the BOSSbase 1.01 dataset [33], experiments are applied on the most recent steganographic techniques: WOW [35], S-UNIWARD [36] and HILL [37], with embedding rates of 0.1, 0.2, 0.3 and 0.4 bpp (*i.e.,* bits per pixel) for WOW and HILL and rates of 0.2, 0.3, 0.4 and 0.5 bpp for the S-UNIWARD. whereas, for ImageNet dataset, experiments are applied on WOW [35] and S-UNIWARD [36], with embedding rates of 0.3 and 0.4 bpp. In all experiments, each dataset is split into 70% for training phase, 10% for validation phase, and 20% for testing phase.

6.3. **Results and Discussion.**

6.3.1. *Results of the Competing Approaches w.r.t. Detection Error.* In the experiments, the proposed approach is trained and tested using the BOSSbase 1.01 stego-images affected by S-UNIWARD steganography algorithm of payloads 0.5, 0.4, 0.3 and 0.2 bpp. The resulting detection errors are reported in Table 1. Then, all competing approaches have been tested by the ImageNet stego-images that are exposed to S-UNIWARD steganography algorithm with payloads 0.4 and 0.3 bpp (*i.e.,* see Table 1).

Similarly, all competing approaches are trained and tested using the BOSSbase stego-images that are affected by both WOW and HILL steganography algorithms with payloads 0.5, 0.4, 0.3 and 0.2 bpp. The detection errors are shown in Table 2 and Table 3, respectively. Then, it is only tested using the ImageNet stego-images that are exposed to both WOW and HILL steganography algorithms with payloads 0.4 and 0.3 bpp (*i.e.,* see Table 2 and Table 3, respectively).

According to the results listed in Tables 1, 2, and 3, it can be noticed that:

- The proposed CNN-based BIS approach outperforms all competing approaches when detecting the S-UNIWARD stego-images with payloads 0.5, 0.4, 0.3 and 0.2 bpp. The proposed approach also outperforms all competing approaches when detecting WOW and HILL stego-images with payloads 0.4, 0.3, 0.2 and 0.1 bpp.
- In the proposed approach, the pre-processing layer has been removed. Instead, a convolutional layer has been added at the beginning of the model. Therefore, the extraction of the stego-noise is improved, thus enhancing the detection accuracy.
- Applying multiple types of activation functions, with respect to the role of each layer and extracted feature maps, has a significant impact on enhancing the accuracy.
- Despite the fact that the depth of the proposed CNN model is lower than that of all competing models, such as YedroudjNet [23] and ZhuNet [25], the proposed CNN model can effectively extract suitable feature maps for image steganalysis issue.

It can be shown that the proposed approach surpasses the competing ones using both image datasets that have been used for verification and validation purposes.

TABLE 1. Detection error of the S-UNIWARD stego-images with different payloads using the BOSSbase [33] dataset for verification, and the ImageNet [34] for validation (*i.e.,* testing only); on the basis of the lower, the better.

Dataset	BOSSbase [33]				ImageNet [34]	
Payload	0.2 bpp	0.3 bpp	0.4 bpp	0.5 bpp	0.3 bpp	0.4 bpp
SRM+EC [7]	32.10	24.95	20.55	16.64	37.70	34.41
GNCNN [10]	37.43	30.62	20.08	17.33	38.52	34.67
IGNCNN [4]	34.38	28.42	22.05	19.32	36.77	33.39
XuNet [13]	39.10	32.84	27.21	18.15	35.18	32.72
YeNet [22]	40.01	35.28	31.21	16.62	33.56	29.63
QianNet [5]	29.42	23.80	17.78	13.96	32.79	27.81
YedroudjNet [23]	36.72	29.61	22.81	18.27	28.31	23.79
ZhuNet [25]	28.59	19.74	15.32	9.56	26.57	22.73
Proposed	**26.72**	**17.91**	**13.47**	**7.94**	**19.86**	**16.22**

TABLE 2. Detection error of the WOW stego-images with different payloads using the BOSSbase [33] dataset for verification, and the ImageNet [34] for validation (*i.e.,* testing only); on the basis of the lower, the better.

Dataset	BOSSbase [33]				ImageNet [34]	
Payload	0.2 bpp	0.3 bpp	0.4 bpp	0.5 bpp	0.3 bpp	0.4 bpp
SRM+EC [7]	40.25	32.10	24.92	20.67	38.21	34.70
GNCNN [10]	50.02	37.47	27.88	20.28	37.37	34.11
IGNCNN [4]	43.93	34.38	24.87	19.62	35.63	33.83
XuNet [13]	33.60	31.65	20.71	18.15	34.02	31.29
YeNet [22]	32.45	24.35	20.36	17.07	33.15	30.50
QianNet [5]	32.61	26.78	19.69	15.07	32.61	29.43
YedroudjNet [23]	32.15	27.83	19.46	14.18	26.32	24.71
ZhuNet [25]	31.84	23.31	17.11	11.82	24.13	22.94
Proposed	**27.03**	**20.54**	**15.81**	**9.65**	**20.08**	**15.38**

TABLE 3. Detection error of the HILL stego-images with different payloads using the BOSSbase [33] dataset for verification, and the ImageNet [34] for validation (*i.e.,* testing only); on the basis of the lower, the better.

Dataset	BOSSbase [33]				ImageNet [34]	
Payload	0.2 bpp	0.3 bpp	0.4 bpp	0.5 bpp	0.3 bpp	0.4 bpp
SRM+EC [7]	48.49	46.29	43.16	41.46	39.87	37.12
XuNet [13]	41.56	37.60	30.12	20.67	36.51	35.98
YeNet [22]	44.30	39.41	34.92	32.45	33.28	32.27
QianNet [5]	39.78	36.82	25.11	19.60	32.74	31.42
YedroudjNet [23]	40.86	35.66	28.64	23.19	30.61	29.85
ZhuNet [25]	32.29	29.87	23.31	16.83	28.32	26.72
Proposed	**30.09**	**26.55**	**20.84**	**13.92**	**25.91**	**20.13**

6.3.2. *Results of the Competing Approaches w.r.t. the Training Time.* The experimental results of proposed CNN-based approach are compared with results of IGNCNN [4], XuNet [13], YeNet [22], QianNet [5], YedroudjNet [23] and ZhuNet [25] in terms of training time and count of applied learning parameters during the training process, which reflects the complexity of the network used. The training time and the counts of training parameters of the competing approaches are presented in Table 4. Results in Table 4 show that the training time required using the proposed approach is significantly decreases compared to competing ones due to the following reasons:

- The implementation of the proposed approach is based on a hybrid technique of the model and data parallelism.

TABLE 4. The total training time (in hours) and the numbers of training parameters of the competing approaches

	Total training time (H)	**Training parameters**
IGNCNN [4]	3.28	71016
XuNet [13]	2.6	**14661**
YeNet [22]	12.43	107671
QianNet [5]	1.66	35508
YedroudjNet [23]	10.62	412082
ZhuNet [25]	23.76	2828562
Proposed	**1.12**	41474

- The TLU and LReLU activation functions, with lower complexity, have been used rather than the Gaussian activation function having higher complexity.
- The number of leaning parameters used in the proposed BIS approach is lower than that of all competing approaches.

7. **Conclusions.** This paper presents a modified CNN model for enhancing the blind image steganalysis technique. The proposed approach is able to tackle the problems of QianNet [5] steganalysis approach by changing the structure of the CNN with efficient initialization to the first convolutional layer in order to compensate the pre-processing layer, using a cyclical learning rate. In addition, the Gaussian activation function is replaced with the LReLU one. The proposed approach and competing ones are implemented on two GPUs, while the former uses a hybrid model and data parallelism technique in order to reduce the training time required.

Regarding the detection accuracy metric, the proposed approach outperforms the SRM +EC [7], GNCNN [10], IGNCNN [4], XuNet [13], YeNet [22], QianNet [5], YedroudjNet [23] and ZhuNet [25] by an average increase of 5.94%, 8.73%, 7.88%, 11.76%, 13.27%, 8.8%, 8.35% and 1% respectively, in low payloads, with S-UNIWARD steganography -based stego-images. It also outperforms the SRM+EC [7], GNCNN [10], IGNCNN [4], XuNet [13], YeNet [22], QianNet [5] and YedroudjNet [23] by an average increase of 8.21%, 12.615%, 10.41%, 9.89%, 2.27%, 2.36% and 2.69% respectively, in low payloads, when detecting WOW steganography -based stego-images, however, with a comparable detection error compared to the ZhuNet [25]. Finally, the proposed approach outperforms the SRM+EC [7], XuNet [13], YeNet [22], QianNet [5], YedroudjNet [23] and ZhuNet [25] by an average increase of 21.2%, 8.6%, 8.96%, 8.17%, 8.26% and 1.51% respectively, in low payloads, with HILL steganography-based stego-images.

From the training time perspective, the proposed approach outperforms the IGNCNN [4], XuNet [13], YeNet [22], QianNet [5], YedroudjNet [23] and ZhuNet [25] by an average improvement in speed-up factor of 3.93X, 8.76X, 11.1X, 2.32X, 9.48X and 21.2X, respectively.

REFERENCES

[1] J.-S. Pan, H.-C. Huang, L. C. Jain, and Y. Zhao, Recent advances in information hiding and applications. *pringer*, 2013.

[2] H.-C. Huang, Information hiding and applications. *Springer*, vol. 227, 2009.

[3] F.-H. Wang, J.-S. Pan, and L. C. Jain, Digital watermarking techniques, em in Innovations in Digital Watermarking Techniques. Springer, pp. 11-26, 2009.

[4] Y. Qian, J. Dong, W. Wang, and T. Tan, Learning and transferring representations for image steganalysis using convolutional neural network, *in IEEE Intern. Conf. on Image Processing*, pp. 2752-2756, 2016.

[5] Y.L. Qian and J. Dong and W. Wang and T.N. Tan, Feature learning for steganalysis using convolutional neural networks, *Multimedia Tools and Applications*, vol. 77, no. 15, pp. 19633-19657, 2018.

[6] J. Fridrich and J. Kodovsky, Rich models for steganalysis of digital images, *IEEE Trans. on Info. Forensics and Sec.*, vol. 7, no. 3, pp. 868-882, 2012.

[7] J. Kodovsky, J. Fridrich, and V. Holub, Ensemble classifiers for steganalysis of digital media, *IEEE Trans. on Info. Forensics and Sec.*, vol. 7, no. 2, pp. 432-444, 2012.

[8] S. Tan and B. Li, Stacked convolutional auto-encoders for steganalysis of digital images, *in IEEE Signal and Info. Processing Association Annual Summit and Conf.*, pp. 1-4, Dec 2014.

[9] T. Pevny, P. Bas, and J. Fridrich, Steganalysis by subtractive pixel adjacency matrix, *IEEE Trans.on Info. Forensics and Sec.*, vol. 5, no. 2, pp. 215-224, June 2010.

[10] Y. Qian, J. Dong, W. Wang, and T. Tan, Deep learning for steganalysis via convolutional neural networks, *in Media Watermarking, Sec., and Forensics 2015*, Intern. Society for Optics and Photonics. SPIE, vol. 9409, pp. 171-180, 2015.

[11] L. Pibre, J. Pasquet, D. Ienco, and M. Chaumont, Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover source mismatch, *Electronic Imaging*, vol. 2016, no. 8, pp. 1-11, 2016.

[12] V. Nair and G. E. Hinton, Rectified linear units improve restricted Boltzmann machines, *in Proc. of the 27th Intern. Conf. on Machine Learning*, pp. 807-814, June 2010.

[13] G. Xu, H.-Z. Wu, and Y.-Q. Shi, Structural design of convolutional neural networks for steganalysis, *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708-712, May 2016.

[14] C. Gulcehre, M. Moczulski, M. Denil, and Y. Bengio, Noisy activation functions, *in Intern. Conf. on machine learning*, pp. 3059-3068, 2016.

[15] G. Xu, H.-Z. Wu, and Y. Q. Shi, Ensemble of cnns for steganalysis: An empirical study, *in Proceedings of the 4th ACM Workshop on Info. Hiding and Multimedia Sec.*, pp. 103-107, 2016.

[16] E. M. Mustafa, M. A. Elshafey, and M. M. Fouad, Accuracy enhancement of a blind image steganalysis approach using dynamic learning rate-based CNN on GPUs, *in The 10th IEEE Intern. Conf. on Intelligent Data Acquisition and Advanced Computing Sys.: Tech. and Applications*, pp. 28-33, Sep. 2019.

[17] Eslam M. Mustafa and Mohamed A. Elshafey and Mohamed M. Fouad, Enhancing the performance of an image steganalysis approach using variable batch size-based CNN on GPUs, *in The 10th IEEE Intern. Conf. on Intelligent Data Acquisition and Advanced Computing Sys.: Tech. and Applications*, pp. 40-44, Sep. 2019.

[18] Eslam M. Mustafa and Mohamed A. Elshafey and Mohamed M. Fouad, Enhancing the performance of CNN-based blind image steganalysis approach using multi-GPU TESLA P100, *IOP Conf. Series: Materials Science and Engineering*, vol. 610, Oct. 2019.

[19] M. Sharifzadeh, C. Agarwal, M. Aloraini, and D. Schonfeld, Convolutional neural network steganalysis's application to steganography, *IEEE Visual Communications and Image Processing*, pp. 1-4, Dec 2017.

[20] M. Salomon, R. Couturier, C. Guyeux, J.-F. Couchot, and J. M. Bahi, Steganalysis via a convolutional neural network using large convolution filters for embedding process with same stego key: A deep learning approach for telemedicine, *European Research in Telemedicine*, vol. 6, no. 2, pp. 79-92, 2017.

[21] W. Tang, S. Tan, B. Li, and J. Huang, Automatic steganographic distortion learning using a generative adversarial network, *IEEE Signal Proc. Letters*, vol. 24, no. 10, pp. 1547-1551, 2017.

[22] J. Ye, J. Ni, and Y. Yi, Deep learning hierarchical representations for image steganalysis, *IEEE Trans. on Info. Forensics and Sec.*, vol. 12, no. 11, pp. 2545-2557, Nov. 2017.

[23] M. Yedroudj, F. Comby, and M. Chaumont, Yedroudj-net: An efficient CNN for spatial steganalysis, *in IEEE Intern. Conf. on Acoustics, Speech and Signal Processing*, pp. 2092-2096, April 2018.

[24] C. F. Tsang and J. Fridrich, Steganalyzing images of arbitrary size with CNNs, *Electronic Imaging*, vol. 2018, no. 7, pp. 1-8, Jan. 2018.

[25] R. Zhang, F. Zhu, J. Liu, and G. Liu, Efficient feature learning and multi-size image steganalysis based on CNN, *ArXiv*, vol. 2018, no. 1, pp. 1-10, 2018.

[26] F. Chollet, Xception: Deep learning with depthwise separable convolutions, *in IEEE Conf. on Comp. vision and pattern recognition*, pp. 1251-1258, July 2017.

[27] K. He, X. Zhang, S. Ren, and J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904-1916, Sep. 2015.

[28] G. Brooks, Gaussian-based filters for elementary motion detector delay element: Modeling spatio-temporal info. pathways with elementary motion detection, *in IEEE Research and Applications of Photonics In Defense Conf.*, pp. 1-3, Aug 2018.

[29] X. Glorot, A. Bordes, and Y. Bengio, Deep sparse rectifier neural networks, *in Proc. of the 14th Intern. Conf. on Arti*
cial Intelligence and Statistics, vol. 15, no. 1, pp. 315-323, 2011.

[30] L. N. Smith, Cyclical learning rates for training neural networks, *in IEEE Winter Conf. on Applications of Comp. Vision*, pp. 464-472, March 2017.

[31] Y. Dauphin, H. De Vries, and Y. Bengio, Equilibrated adaptive learning rates for non-convex optimization, *in Proc. of The 28th Intern. Conf. on Neural Info. Processing Sys., ser. NIPS'15*, vol. 1, no. 1, pp. 1504-1512. MIT Press, 2015.

[32] Y. Qian, J. Dong, W. Wang, and T. Tan, Feature learning for steganalysis using convolutional neural networks, *Multimedia Tools and Applications*, vol. 77, no. 15, pp. 19633-19657, Aug 2018.

[33] P. Bas, T. Filler, and T. Pevny, Break our steganographic sys.: The ins and outs of organizing boss, *in Info. Hiding*, pp. 59-70, 2011.

[34] S. V. Lab, ImageNet: image database. [Online].Available: https: //www.http://image-net.org/index.

[35] V. Holub and J. Fridrich, Designing steganographic distortion using directional filters, *in IEEE Intern. Workshop on Info. Forensics and Sec.*, pp. 234-239, Dec 2012.

[36] V. Holub and J. Fridrich, Digital image steganography using universal distortion, *in Proc. of the*
rst ACM Workshop on Info. Hiding and Multimedia Sec., pp. 59-68, 2013.

[37] B. Li, M. Wang, J. Huang, and X. Li, A new cost function for spatial image steganography, *in IEEE Intern. Conf. on Image Processing*, pp. 4206-4210, Oct 2014.