Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 9 Issue 9, September - 2023

An Enhanced Convolutional Neural Network Model for Image Steganalysis

Uwadia Anthony O.
Department of Information and Communication Engineering
Elizade University, Ilara-mokin, Ondo State
anthony.uwadia@elizadeuniversity.edu.ng, engr.anthonyuwadia@gmail.com

***Abstract*—CNN exciting level of accuracy in stego image classification has given rise to a new state-of-the-art. The wide use of images to encrypt Information has heightened the need for a resourceful classifier like Convolution Neural Network (CNN) to detect whether an image is what it appears to be. A well-ordered CNN model will augment its practical deployment and advance scientific understanding in image steganalysis.**

In this research, we proposed a CNN classifier-based JPEG color image steganalysis to discriminate stego images from their cover counterparts. Structurally, a bi-segmental CNN model was used to implement the deep learning aptitude of CNN, while the Adaptive Momentum (Adam) optimization technique was used to enhance its performance. The model is structured to learn important image features while avoiding data loss for decent predictive classification accuracy. The experimental results of the research study show the model's suitability in terms of its ability to perform image steganalysis with no prior knowledge of the embedding algorithm and rate.

Keywords— component; Dataset; Convolution Neural Network (CNN); Adaptive Momentum (AdaM); Steganalysis ; Image Steganalysis

I. INTRODUCTION

Data or Information is concealed in a cover media to make it indistinguishable from unauthorized personnel. Steganalysis is the art and science of detecting whether a given digital medium contains hidden data. Image steganalysis is the reverse process of image steganography, determining whether an image contains an embedded secret message. Its goal is to identify suspected image packages and distinguish between those with payloads from ordinary ones. It begins by identifying the artifacts in the suspect image stego file, which has formed due to a message being embedded. The interrupted data in cryptanalysis often contains a message. At the same time, the steganalysis system presents a bundle of suspect data files that cannot be differentiated from each other in terms of those files containing the payload. Image steganalysis methods usually relate to image repositories to extract images with steganographic alterations with high-pass filters (HPF) to detect images that contain artifacts in addition to their original contents. Researchers in this field carefully consider the image statistical properties that cannot be traced to stego-system, paving the way to find out whether an image is in its ordinary form or not. Thus, a successful steganalysis system can be identify as long as it has higher degree of probability to predict the images that are stego as stego and the cover as cover [15].

II. RELATED WORKS

Steganalysis has applications in several fields, such as extracting the stego message, deciphering suspicious hidden messages, and evaluating the robustness of existing steganography techniques [1] [14]. [2] developed a JPEG-Phase-Aware Convolutional Neural Network (PNet) for the steganalysis of JPEG images. They aimed to improve detection accuracy by incorporating JPEG awareness into CNN architecture alongside traditional features. Their methodology introduced a phase-split layer and integrated phase awareness into the network early layers. Experiments on BOSSbase and BOWS2 datasets demonstrated PNet’s slightly better performance than VNet(an existing model), though it required more computation and training time. Their research was motivated by the realization that the modern steganographic algorithm traditionally relied on the JPEG phase's feature awareness. [3] proposed a JPEG steganalysis based on DenseNET. They worked on a 32-layer Convolutional Neural Network (CNN) to improve the efficiency of preprocessing and reusing the feature by concatenating all features from the previous layer with the same feature map size. Their approach included high-pass filtering and trainable truncating processes. Their research was motivated by the need to use shared feature and bottleneck layers to improve the feature propagation further and reduce the CNN model parameter using resemblance architecture (CNN-SCA-GFR). While it enhances spatial steganalysis, it didn’t perform well for JPEG steganalysis and wasn’t suited for selection channel awareness. [4] focused on improving CNN-based steganalysis performance by augmenting a small learning set through data enrichment. They used virtual augmentation to expand their initial 4000-pair database, significantly improving performance. However, the research didn't deeply explore generalization or cover-source mismatch issues. [5] Their research methodology involved a five-block convolutional module with different kernel sizes and truncation functions. Batch normalization was used to standardize feature distributions. The research aimed to create a CNN for spatial steganalysis which is efficient, utilizing transfer learning or virtual augmentation, and achieved a model that is not sensitive to hyperparameter initialization. [6] developed an advanced CNN-based steganalysis method to address fixed-size image issues in existing algorithms. They introduced a modified CNN architecture with a small convolution kernel, batch normalization, and ReLU activation.They aimed to create a model to analyze arbitrary image sizes and improve signal-to-noise ratios. Their model effectively handled diverse image sizes using a spatial pyramid pooling (SPP) module. Data augmentation and SPP improved performance, but the research noted limitations in data augmentation methods that could harm pixel correlation and network performance.[7] developed the WISERNet model for the steganalysis of color images. They used channel-wise convolution to make the network more "hidden pattern-aware" in the lower convolutional layer, focusing on detecting stego noise while suppressing content. Their model performed well against CMD-C, a recent steganographic algorithm for color images, but was relatively shallow and unsuitable for large JPEG image databases.[8] developed a CNN-based steganalysis method for ternary classification, aiming to distinguish between cover, WOW stego, and UNIWARD stego images in a single classifier. They explored combining binary classifiers and transfer learning. They discovered that adding more convolutional layers improved classification rates up to a point, indicating the need for careful depth adjustment for ternary classification. [9] used a joint domain detection mechanism and a nonlinear detection mechanism for image steganalysis based on CNN to improve the detection of adaptive steganographic algorithms with low embedding rates. Their research model consists of 10 network layers, preprocessing layers with convolution kernels, batch normalization, and ReLU and TLU nonlinear activation functions. Their research demonstrates that their model is more effective than other spatial domain steganalysis methods in extracting diverse steganography information, but it comes with a higher time complexity due to its complexity. [10] developed a lightweight convolutional neural network called IAS-CNN for image adaptive steganalysis. Their research approach included a preprocessing layer for residual calculation, feature extraction, and classification layers. The preprocessing layer used SRM filters to extract residuals, and the feature extraction layer employed convolutional layers with ReLU activation functions and average pooling. IAS-CNN outperformed deeper networks like ZhuNet and YeNet in terms of efficiency, particularly with higher payloads, but showed lower performance with low payloads. [11] In their approach, they proposed an image steganalysis which focused on utilizing diverse filter modules (DFMs) and squeeze-and-excitation modules (SEMs) to capture embedding artifacts. Their research approach included seven (7) convolutional layers with different kernel sizes, batch normalization, ReLU activation, and average pooling. DFMs processed feature maps in parallel with 3 X 3 and 5 X 5 kernels, then integrated through a 1 X 1 convolutional layer. SEMs were used to learn feature weights. DFSE-Net outperformed traditional methods in steganalysis, particularly for the WOW and S-UNIWARD steganographic algorithms at 0.2 and 0.4 bpp payloads. However, due to computing limitations, it worked with 256 X 256 input images.

The shortcomings of the reviewed works are identified in the suitability of grayscale image steganalysis due to its data embedding restrictions such as 8-bit image data holding capacity which is a small volume and grey images format usually BMP file which are not web friendly with current RGB images. The colour image steganalysis model provided is not equally appropriate to carry out a blind steganalysis method that has no prior knowledge of the embedding algorithm since the universal steganalysis approach is an appropriate method of exploring the power of CNN classifier with a learning-based strategy that involves a training stage and a testing stage.

Stimulated by [7], this research study proposes an Image steganalysis model with respect to JPEG image dataset using Adaptive momentum optimizer to explore the depth power of CNN with careful consideration of the choice of filter size and number, the CNN layers arrangement in order to both knob the requirement for sufficient feature extraction, adding effective regularization effect to circumvent overfitting and produce a workable minimum number of parameters at the final feature maps.

A. *CNN based approach*

CNN comprises one or several layers trailed by some other layers, including fully connected. Regardless of the types of layers, they are made of units with learnable weights and biases. Each unit gets inputs from a few units of the preceding layer, carries out a dot product with weights, and optionally computes a nonlinear point-wise activation function. The contextual idea of the convolutional layers is the ability to extract sets of smaller feature maps with kernels from two-dimensional input data like images to produce the final feature map and then process by the fully connected part that eventually carries out the classification to give a better representation of the original data.

B. *Image Steganalysis*

Among various algorithms such as Support Vector Machine (SVM), Naïve Bayes etc based image steeganlysis, the CNN based image steganalysis have recently shown superiority in discerning the stego image from its cover counterparts [16], and it outperforms conventional methods due to its capability to learn the input features with deep layers of neuron structures without manual supervision [17]. CNN image steganalytic algorithm operates in two steps to solve various problems. The objective of the first step is to capture ample data on the input images by striding and mapping through the set of features.

The second step aims at classifying the images into either stego or cover in line with the extracted values of certain features, which can be used to provide a decision-making system.

Precisely, there are two steps steganalysis machine can be summarized into Training and Testing phases [1]. The training phase extracts features from the input data to form a feature matrix of size $n \times m$ through mapping. This feature matrix is then used to train the CNN classifier. The testing phase provides the platform by which the trained CNN can be used to perform image classification into stego and cover image.

III. METHOD

The proposed CNN image steganalysis required four phases: The Data Collection phase is the first phase in this research study. The preprocessing phase is the second. The feature extraction or CNN training phase is the third phase, and the CNN testing is the final phase. As illustrated in Figure 1

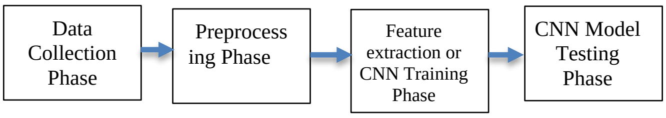

Fig. 1. *CNN image Steganalysis Model flow chart*

A. Data Collection

The dataset is acquired from the github.com image repository named IStego100k, initially crawled from Unsplash5 copyright-free photography website using API provided by the Unsplash website and prepared for public use by image steganalysis researchers. The selected image is 4187, originally defined as 2631 cover images and 1551 stego images. The entire dataset was resized to 70x70 dimensional space to make it suitable to train and test the model since the fully connected layer of the CNNs required a fixed image size. The datasets were divided into training and test sets. The training set takes the larger percentage of the data (70%), while the remaining 30% is apportioned to the test set. The training sample constituted 2930 images, while the test sample constituted 1257 images of 3-dimensional shape. 2,930 images were randomly selected, containing stego and cover images with no consideration of the embedding algorithm and rate to reflect the real-world image steganalysis when exposing images to know how they appear. The training set has 1089 stego images and 1841 cover images. The test set has 790 cover images and 467 stego images, as shown in Table 1.

TABLE 1. IMAGE DATASET DISTRIBUTION

Total Images	***Training Set (2,930) – 70%***	***Testing Set (1,257) – 30%***
Cover Images	1,089	790
Stego Images	1,841	467

B. Image Preprocessing

Data preprocessing and augmentation were performed before inputting the data into the CNN model for training. In this phase, a high pass filter is used on the imputed datasets to make the model concentrate on the embedded data rather than the image contents. The filter emphasizes fine features in the image; small and faint details are greatly exaggerated. After that, the filtered images are normalized using the min-max method as described in (1) as

$$x' = \frac{x - mean(x)}{\max(x) - \min(x)} \quad (1)$$

where x' and x are the input-image values. The dataset becomes suitable for training and classification since the image contents are stronger than the implanted signals, which is the noise content added to the image.

C. CNN Training

In Image Steganalysis, the key to successful classification is to compare the artifacts introduced by steganography instead of the image contents. A multilayer architecture (multi-task learning) was employed to amplify the important aspects of the input layers while becoming more robust to less significant variations. The proposed architecture comprises six (6) convolutional blocks and a fully connected layer. Each of the first three (3) convolutional blocks is made up of four (4) components: The Convolutional layer with filter size (5 x 5), Concatenated Rectifier Linear Unit (CReLU) activation function, Pooling layer, and batch normalization. While the last three (3) convolutional blocks with filters at each convolutional layer perform the convolution process of input-image transformation into the output feature maps. These latter three (3) convolutional segment contains three (3) components as the previous blocks except the pooling layer, as illustrated in Figure 2 [12]

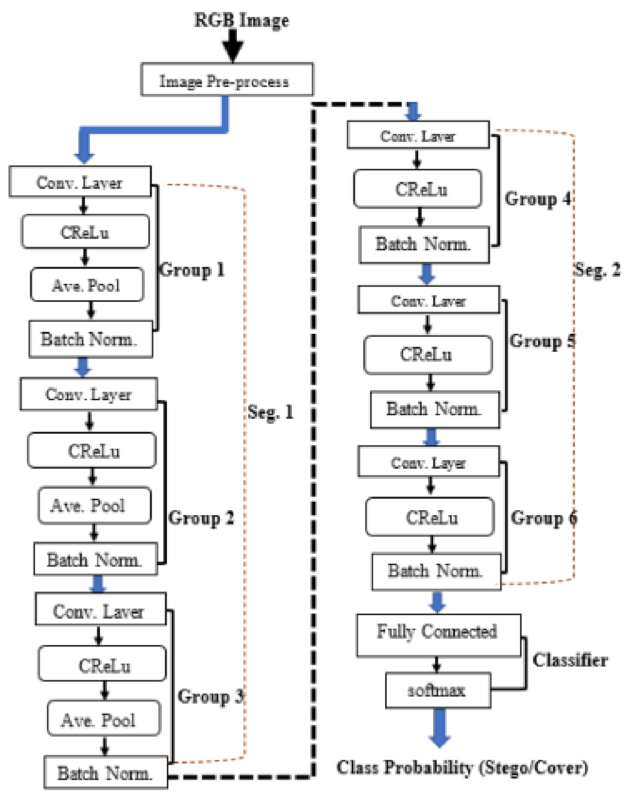

Fig. 2. *Proposed Architecture for image steganalysis. Adapted [12]*

D. Convolutional Layers

In this research study, the proposed six (6) convolutional blocks are divided into two major segments (3) convolution blocks in each segment). The first segment contains a (5x5) filter size for each convolutional block and (3x3) for each convolution block at the second segment. This filter size arrangement ensures more image feature extraction and minimizes the output number of parameters at the final feature map. Full data subsampling was not applied to the proposed model because it can lead to noisy objectives. Due to this effect, it allows us to implement Adam optimization technique to enhance the model's performance. In the convolutional layer, (2) represents the result in layer l of the convolution with the k-th kernel. From the input image, the weighting matrix,

$$C^{kl} = \sum_{m=1}^{k^{l-1}} (W^{kl} * F^{m(l-1)}) \qquad (2)$$

Where * denotes the convolution product. K^{l-1} is the number of kernels in the preceding layer, and $F^{m(l-1)}$ is the m-th final feature map produced by layer *l*. For the first convolutional layer, i.e when *l* =1, $K^{l-1} = K^0 = 1$, and $F^{1(l-1)} = F^{1(0)} = 1$ which is the input-image. W^{kl} is the size of the matrix filter, which implicitly defines the size of the region of interest. Stride, S, and padding, P are employed to influence the output size of convolutional operations, the square input image/map $F^{m(l-1)}$ an output C^{kl} such as that in (3)

$$dim(C^{kl}) = \frac{(dim\ (F^{m(l-1)} - dim(W^{kl} + 2\ x\ P)}{S+1} \qquad (3)$$

To keep C^{kl} the same size as the input data $F^{m(l-1)}$, $dim(C^{kl}) = (dim\left(F^{m(l-1)}\right)$, the stride and padding values must be set to 1 and $\frac{dim(W^{kl})}{2}$, respectively

E. Activation Function

In this step, instead of using Rectified Linear Unit (ReLU), we considered a variation of ReLU called Concatenated Rectified Linear Unit (CReLU), which leads to better performance while reducing the number of parameters to be learned by removing redundancies. Equation 3 describes how the activation function can be modeled with various possible choices with each convolutional layer C^{kl} and an activation function f^{kl} (CReLU) to ensure its non-linearity as well as reduce the number of parameters by half when compared with the ReLU activation function when used in the model

$$f(x) = \frac{e^{-x^2}}{\sigma^2} \qquad (4)$$

A bias b^{kl} in (4) for each kernel will yield parameter optimization in layer 1. Ultimately resulting in A^{kl} with f^{kl} as the function for each kernel, as shown in (5)

$$A^{kl} = f^{kl}(C^{kl} + b^{kl}) \qquad (5)$$

F. Pooling Layer

To preserve enough feature information after subsampling, we applied an average pooling operation in the first three (3) convolutional layers. We employed an average pooling over max pooling support downsampling with better abstraction, smaller feature maps, and wider receptive fields. During the pooling stage, 2D maps are down-sampled into $Pl\ x\ Pl$ regions, replaced by mean or maximum values following (6), (7) and (8)

$$F^{kl} = pooling\ (A^{kl}) = pooling\ (f^{kl}\ (C^{kl} + b^{kl})) \qquad (6)$$

$$F^{kl} = pooling\ \{f^{kl}\ \{\textstyle\sum_{m=1}^{k^{l-1}} (W^{kl} * F^{m(l-1)}) + b^{kl}\}\} \qquad (7)$$

$$F^{kl} = pooling\ (\ f^{kl}\ (\ W^{kl} * I + b^{kl})) \qquad (8)$$

Where I is the original input image

Some datasets are prone to overfitting. However, batch normalization was applied in all convolution blocks to mitigate this effect by smoothing the loss function, stabilizing the learning process, and reducing the training epochs; by so doing, the training efficiency and generalization effect are improved. The fully connected layer evaluates features from the convolution layers and generates an N-dimensional probability vector where N is the number of the classification targets. The proposed architecture used is one fully connected layer with drop-out to combat overfitting in the model. (9) represents the softmax function's final classification probability, which was used to generate the result to discriminate between the cover images and their stego counterparts.

$$\Pr(Y_i = k)\ \frac{e^{\beta_k . X_i}}{\sum_{k=1}^{k} e^{\beta_k . X_i}} \qquad (9)$$

Where Y_i is the probability set, β_k is the set of regression coefficients associated with outcome k, and X_i is the set of explanatory variables associated with observation, i

The Adaptive Momentum (AdaM) optimization technique, as shown in (10) – (13), was adopted to estimate the adaptive learning rate for all parameters involved in the training of gradients, which in turn will enhance the system's performance

$$x_t = \delta_1 * x_{t-1} - (1 - \delta_1) * g_t \quad (10)$$

$$y_t = \delta_2 * y_{t-1} - (1 - \delta_2) * g_t^2 \quad (11)$$

$$\Delta\omega t = -\mu \frac{x_t}{\sqrt{y_t + \epsilon}} * g_t \quad (12)$$

$$\omega_{t+1} = \omega_t + \Delta\omega_t \quad (13)$$

µ denotes the initial learning rate, g_t is the gradient at time t along ω_t, x_t is the exponential average of the gradient along ω_t, y_t is the exponential average of squares of the gradient along ω_t and δ_1, δ_2 are the hyperparameters. The evaluations such as Accuracy (ACC), Precision, and Recall were performed based on the Confusion matrix performed using (14) - (16). [13]

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (14)$$

$$Precision = \frac{TP}{TP+FP} \quad (15)$$

$$Recall = \frac{TP}{TP+FN} \quad (16)$$

Where TP is True Positive, TN is True negative, FN is False Negative, and FP is False Positive.

Model Development

The dataset contains 2,930 images randomly selected from github.com, including stego and cover images, without considering the embedding algorithm and rate to reflect the real-world steganalysis conditions for accurate representation. Before the model training, data preprocessing and augmentation were conducted, and the entire dataset was resized to 70 by 70 dimensions to accommodate the requirements of the fully connected layer in CN.

The learning rate was set to 1.0 in training the model with 20 training epochs. The choice of this value creates a hybrid of oscillation between high and low learning rates to sidestep any difficulties encountered by both. The performance of the steganalysis model is evaluated based on its predictive accuracy using three (3) different set-ups to depict the effect of each of these settings on the model structural arrangement. Figures 3, 4, and 5 show the three(3) different scenarios based on their predictive accuracy: The training and validation accuracy without parameter tuning, the training and validation accuracy with parameter tuning, and the training and validation accuracy graph with parameter tuning and discrete wavelength transform. Tables 2 and 3 show the Data Plot on Training and validation and the Predicted outcomes of different epochs for each graph instance. Table 4 shows an assessment of the proposed model with the existing models.

TABLE 2. DATA PLOT ON TRAINING AND VALIDATION DATA

Epoch	Train-loss	T-accuracy	Valid-loss	V-Accuracy	Time
1	0.8097	0.3696	0.7397	0.3715	00.88
2	0.7223	0.3913	0.7012	0.3715	00.87
3	0.6977	0.4130	0.6782	0.6385	00.86
4	0.6760	0.6304	0.6640	0.6385	00.89
5	0.6847	0.5870	0.6650	0.6385	00.87
6	0.6834	0.5870	0.6647	0.6385	00.89
7	0.6459	0.6630	0.6600	0.6385	00.87
8	0.6429	0.6413	0.6608	0.6385	00.87

TABLE 3. PREDICTED OUTCOMES OF DIFFERENT EPOCHS FOR EACH GRAPH INSTANCE

Graph Chart	Epochs	Training Accuracy	Validation Accuracy
Graph results without parameter tunning	1	62.00%	56.30%
	2	61.80%	58.00%
	3	62.20%	58.20%
	4	62.40%	57.80%
	5	63.80%	57.70%
	6	64.00%	57.20%
	7	64.40%	57.00%
	8	64.00%	56.80%
Graph results with parameter tunning	1	20.00%	15.00%
	2	42.00%	63.00%
	3	63.00%	63.00%
	4	57.50%	63.00%
	5	57.50%	63.00%
	6	63.00%	63.00%
	7	65.00%	63.00%
Graph results with parameter tunning and DWT	1	48.00%	63.00%
	2	68.00%	63.00%
	3	57.00%	63.00%
	4	53.00%	63.00%
	5	63.00%	63.00%
	6	62.00%	63.00%
	7	65.00%	63.00%

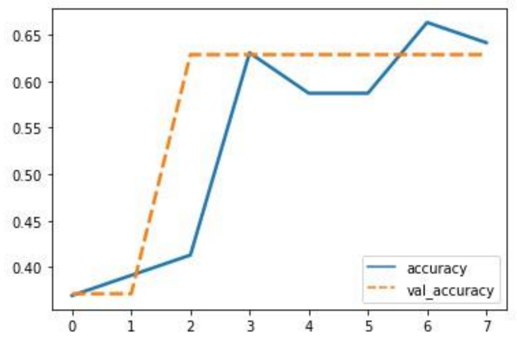

Fig. 3. *Graph for Training and validation with parameter tunning*

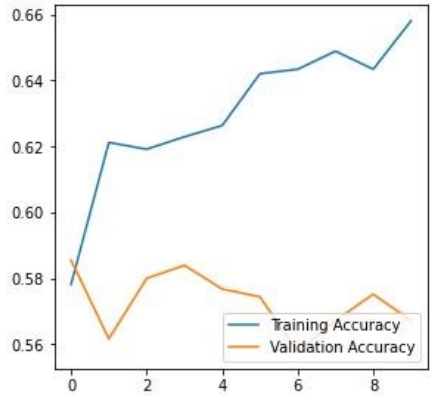

Fig. 4. *Graph for Training and validation accuracy without parameter tunning*

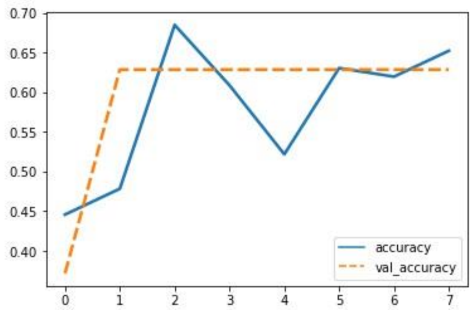

Fig. 5. *Training and validation accuracy graph with parameter tuning and discrete wavelength transform*

TABLE 4. AN ASSESSMENT OF THE PROPOSED MODEL WITH THE EXISTING MODELS

Model Name with Dataset	Data Embedding rate	Accuracy
WiserNet Model Accuracy (BOSS- PPG, AHD -LAN)	0.2bpc	0.725
	0.4bpc	0.8411
Ye's Model Accuracy (BOSS- PPG, AHD -LAN)	0.2bpc	0.6844
	0.4bpc	0.8031
Xu's Model Accuracy (BOSS- PPG, AHD -LAN)	0.2bpc	0.7055
	0.4bpc	0.8118
Proposed Model Accuracy (Github image dataset)	Not specified	0.6285

CONCLUSION

This work explores the potential of CNN to create an effective image steganalysis model, which emphasizes a careful selection of filter size and number and the arrangement of CNN layers to balance adequate feature extraction with minimizing parameters in the final feature map. In addition to the high pass filter used to preprocess the input image data, we employed multilayer architectures (multi-task learning) to amplify the important aspects of the input layers while becoming more robust to less significant variations. With this, the system learns to be implicitly aware of the level of noise it faces. The resulting architecture leads to a better model for the main task since it allows commonality among the tasks.

The proposed CNN-based image steganalysis system holds promise for adoption in Information Security Systems, verifying the authenticity of images in social networks as either original or steganographic versions.

REFERENCES

[1] D. A. Shehab, and M. J. Alhaddad, “Comprehensive Survey of Multimedia Steganalysis: Techniques, Evaluations, and Trends in Future Research,” *Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia,* vol. 14, no. 1, 2022, pp. 117.

[2] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich, “JPEG-Phase-Aware Convolutional Neural Network for Steganalysis of JPEG Images,” *In Proceedings of IH & MMSec’17*, Philadelphia, PA, USA, 2017, pp. 75–84.

[3] J. Yang, Y.-Q. Shi, E. K. Wong, and X. Kang, “JPEG Steganalysis Based on DenseNet,” .Retrieved from http://arxiv.org/abs/1711.09335. 2017, pp. 1-7

[4] M. Yedroudj, M. Chaumont and F. Comby,”How to augment a small learning set for Improving the performances of a CNN-based steganalyzer.?” 2018, *IST Int. Symp. Electron. Imaging Science Technology.*

[5] M. Yedroudj, F. Comby and M. Chaumont, “Yedroudj-Net: An Efficient CNN for Spatial Steganalysis,” *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, April 2018, pp. 2092–2096.

[6] R. Zhang, F. Zhu, J. Liu and G. Liu, “Efficient feature learning and multi-size image steganalysis based on *CNN,”* 2018, vol. 2, pp 1–10, Retrieved from http://arxiv.org/abs/1807.11428

[7] J. Zeng, S. Tan, G. Liu, B. Li and J. Huang, “WISERNet: Wider Separate-Then-Reunion Network for Steganalysis of Color Images,” *IEEE Transactions on Information Forensics and Security*, Images: A Review. *Journal of Artificial Intelligence,* 2019, vol. 10, pp. 1-21

[8] S. Kang, H. Park and J. Park, “CNN-based ternary classification for image steganalysis," *Electronics (Switzerland)*, 2019, vol. 8, no. 11, pp 1–15.

[9] Z. Wang, M. Chen and Y. Yang, “Joint multi-domain feature learning for image Steganalysis

based on CNN," *EURASIP Journal on Image and Video Processing*. Springer open. 2022, vol. 7.

[10] Z. Jin, Y. Yang, Y. Chen and Y. Chen, "IAS-CNN : Image adaptive steganalysis via convolutional neural network combined with selection channel," *International Journal of Distributed Sensor Networks.* 2020, vol. 16, no. 3, pp. 3.

[11] X. Yan, Y. Lu and S. Wang, "Image Steganalysis via Diverse Filters and Squeeze-and Excitation Convolutional Neural Network," *Mathematics*, 2021, vol. 9, pp. 189.

[12] M. Sharifzadeh, C. Agarwal, M. Aloraini, and D. Schonfeld, "Convolutional neural network steganalysisis application to steganography," *2017 IEEE Visual Communications and Image Processing*, VCIP 2017, January 2017, pp 1–4. 2018.

[13] L. Wei, L. Kong, Z. Liu, Z. Yang, and H. Zhang, "A Low-Complexity Accurate Ranging Algorithm for a Switch Machine Working Component Based on the Mask RCNN". Applied Sciences, 2023. vol. 13, no. 16, pp. 9424.

[14] S. Rahman, J. Uddin, H. Hussain, J. Salman, J. Salman, I. Khan and M. Shabir, "Multi Perspectives Steganography Algorithm for Color Images on Multiple-Formats." Sustainability, 2023. vol. 15, no. 5, pp. 4252.

[15] M. Bouzegza, A. Belatreche, A. Bouridane, and M. Tounsi, "A comprehensive review of video steganalysis," *IET image processing*, 2022, vol. 16, no. 13, pp. 3407-3425.

[16] D. S. Sakkeena, S. Murugavalli, L. Jabasheela, and V. Anitha, "Ceaseless steganographic approaches in machine learning," *Meas. Sensors*, 2023, vol. 25, December 2022, pp. 100622.

[17] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Computer Science*, 2021. vol. 2, no. 6, pp 1–20.