

Steganalysis for Secret Message Length Estimation using GBRAS Net Regressor

Ratih Kartika Dewi

Sch. of Electrical Eng. and Informatics
Bandung Institute of Technology
Bandung, Indonesia
33222014@std.stei.itb.ac.id

Rinaldi Munir

Sch. of Electrical Eng. and Informatics
Bandung Institute of Technology
Bandung, Indonesia
rinaldi@staff.stei.itb.ac.id

Nugraha Priya Utama

Sch. of Electrical Eng. and Informatics
Bandung Institute of Technology
Bandung, Indonesia
utama@staff.stei.itb.ac.id

Abstract—The main objective of steganalysis is to predict whether a suspect image is a cover image or a stego image. After predicting the presence of a secret message, further steganalysis research continues by estimating the length of the secret message. Research on estimating the length of secret messages aims to validate the existence of secret messages by providing measurable evidence that a digital medium, particularly an image, contains a secret message of a certain length. Estimation of the length of secret messages embedded using the S-Uniward adaptive steganography algorithm in previous works, which utilized a pretrained ResNet-50, shows high MAE values. This performance indicates the need for improvements in the deep learning regressor architecture. Therefore, this study proposes the development of GBRAS Net for estimating the length of secret messages by modifying the classification layer into a regression layer. The modification involves replacing the Softmax loss function with Mean Squared Error (MSE) and using continuous values as a substitute for payload class labels. This study aims to develop a predictive model to estimate the length of secret messages using the GBRAS Net regressor on the Bossbase 1.01 dataset. The proposed model shows the lowest MSE (0.0182), RMSE (0.1349), and MAE (0.1064) values among ResNet 50, VGG-16, and Ye Net regressor.

Index Terms—image steganalysis, quantitative, regression, GBRAS Net.

I. INTRODUCTION

The main objective of steganalysis is to predict whether a suspect image is a cover image (does not contain a secret message) or a stego image (contains a secret message). Research on predicting the existence of secret messages in a digital image is called binary steganalysis [1]. After predicting the presence of a secret message, further steganalysis research continues with estimating the length of the secret message, predicting the type of insertion algorithm, the key used, and extracting the secret message [2]. Research on estimating the length of the secret message is referred to as quantitative steganalysis. Quantitative steganalysis aims to validate the presence of a secret message by providing measurable evidence that a secret message of a certain length exists within an image. If the estimation algorithm returns a length close to zero, there is a possibility that the prediction result of binary steganalysis is a false positive. Conversely, if the estimation yields a significant length, this serves as strong evidence that

there is indeed a hidden secret message within an image. This means that quantitative steganalysis not only detects the presence or absence of hidden messages but also estimates the size or length of the message embedded in digital media, particularly images [3]–[5].

The quantitative steganalysis methods utilize machine learning [4] and deep learning [3]. While research on binary steganalysis using machine learning is conducted using classification methods [6], [7], research on message length using machine learning can be conducted in two ways: the first is multi-class classification, and the second is using regression methods. Multi-class classification can be performed if the training and testing data are embedded with messages whose payload classes are already determined [8], for example, five classes for payload measurement: very low (<0.2 bpp), low (0.2 bpp – 0.4 bpp), medium (0.4 bpp - 0.6 bpp), high (0.6 bpp - 0.8 bpp), and very high (>0.8 bpp). For measuring message length where the payload class is unknown (data in the form of images with continuous payload values), regression methods are used to predict the payload size of an image, as described in [4], which predicts message length using a rich model for feature extraction [9] and an ensemble regressor for regression. The rich model is used as feature extraction for both binary and quantitative steganalysis. The classification stage in binary steganalysis, which is a decision tree-based ensemble classifier, is changed to regression with a regression tree-based ensemble regressor.

Machine learning methods for steganalysis are increasingly shifting toward deep learning [10]. Current deep learning research on message length [3], [5] uses regression (deep learning regressor) to predict message length with payloads ranging from 0.1 to 0.6 bpp. Message length estimation was performed by adapting the binary steganalysis CNN that had the best accuracy at that time, namely Ye Net [11], by changing the classification layer to a regression layer. Research [3] uses the S-Uniward steganography algorithm, while [5] uses LSB. According to [12] quantitative steganalysis can be done using CNN-Long Short Term Memory on the UCID image database. Quantitative steganalysis using deep learning methods then evolved using a pretrained network [13]. The pretrained network used is ResNet-50 [14]. The limitation of the study [13] lies in performance. For estimating the

length of the secret message embedded using the adaptive steganography algorithm S-Uniward, the MAE value of 4.46 is considered a high error value. However, this value is better than using VGG-16 [15], which produces an MAE value of 9.45.

Feature extraction in binary steganalysis affects quantitative steganalysis [4]. Pretrained networks such as Le Net, Alex Net, VGG Net, Google Net, and ResNet-50 do not perform well for binary steganalysis [5], while specifically designed CNNs for binary steganalysis achieve higher accuracy than pretrained networks [16]. If the performance of binary steganalysis affects the performance of quantitative steganalysis, then improvements to the CNN architecture in binary steganalysis will also improve the performance of quantitative steganalysis. Therefore, this study proposes a deep learning regressor model for quantitative steganalysis that is capable of estimating the length of secret messages in images resulting from the insertion of the S-Uniward steganography algorithm by utilizing the feature extraction performance of binary steganalysis CNN. The binary steganalysis CNN used in this study is GBRAS Net. GBRAS Net was chosen because it has superior performance compared to Ye Net for binary steganalysis [16]. This is based on the fact that feature extraction with high accuracy in binary steganalysis will also improve the accuracy of quantitative steganalysis [3]–[5]. The classification layer was then converted into a regression layer with three fully connected layers, accompanied by replacing the softmax loss function with Mean Squared Error (MSE) and replacing the payload class label with a continuous value. In summary, the contributions of this research are as follows:

- Developed new stego data for quantitative steganalysis research, as the stego data used in previous studies [3], [13] are not publicly accessible.
- Adapted the GBRAS Net CNN architecture [16] for feature extraction, then proceeding with regression to estimate the length of the secret message.

II. LITERATURE REVIEW

The literature review discussed in this study is previous works in quantitative steganalysis. Several deep learning regressor used as reference in this study explained in detail in the next subsection.

A. Quantitative Steganalysis

In general, there is a difference in pattern between the cover image and the stego image resulting from insertion using the S-Uniward adaptive steganography algorithm, namely the presence of noise or residue in the stego image in areas with complex textures as in Fig.1, so that the message insertion is not evenly distributed throughout the image. This is why the algorithm is named the adaptive steganography algorithm, as it embeds the secret message in specific parts of the image (complex textured areas) to minimize suspicion. This differs from the Least Significant Bit (LSB) method (a non-adaptive steganography algorithm), which embeds secret messages into images without selecting which pixels to embed the

messages into [17]. This characteristic distinguishes adaptive and non-adaptive steganography algorithms, so that current deep learning-based binary steganalysis research is designed to detect secret messages embedded with adaptive steganography algorithms but is not tested on non-adaptive steganography algorithms such as LSB. The message length (payload) is

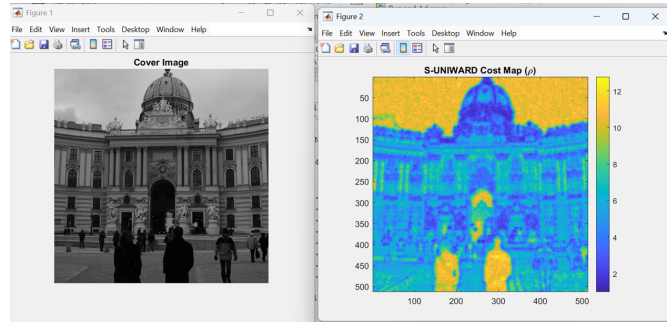


Fig. 1. Costmap S-Uniward [17]

calculated based on 1, which is by dividing the number of secret message bits by the total number of pixels in the cover image. If 1 secret message bit is inserted into each pixel of the cover image, it will result in a value of 1 bpp (bits per pixel).

$$bpp = \frac{\text{number of secret message bits}}{\text{number of pixels in the cover image}} \quad (1)$$

Quantitative steganalysis estimates the length of secret messages using various methods, the latest of which is a deep learning regressor. Deep learning (DL) regressors can be adapted from CNN architectures for binary steganalysis by changing the classification layer to a regression layer. This change is accompanied by a change in the softmax loss function from accuracy to Mean Squared Error (MSE) and the replacement of payload class labels, which were previously categorical (value 0 for cover images and 1 for stego images), with continuous values. The next subsection will discuss Convolutional Neural Network (CNN) for quantitative steganalysis.

B. Ye Net

Ye Net [11] is a CNN based binary steganalysis that has three main components, namely the preprocessing module, convolution module, and regression module, as shown in Fig. 2. The preprocessing module prepares image data so that it is suitable for processing in the next stage. In the Ye Net preprocessing stage, input images measuring $1 \times 256 \times 256$ are processed using an SRM filter. The convolution module functions to extract important features from the image. It consists of several blocks, with each block representing one convolution stage. Average pooling takes the average value to reduce the data dimension while retaining important features. This process is repeated several times, with each convolution layer extracting increasingly complex features. Ye Net regressor [3] is adapted from Ye Net architectures for binary steganalysis by changing the classification layer to a regression

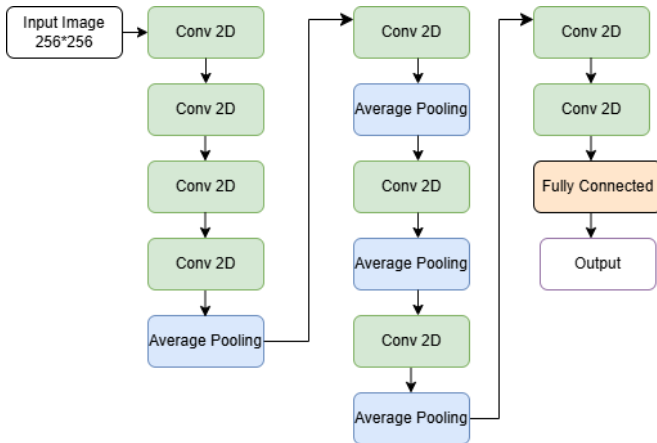


Fig. 2. Ye Net [11]

layer. This change is accompanied by a change in the softmax loss function from accuracy to Mean Squared Error (MSE) and the replacement of payload class labels with a continuous value. The deep learning regressor architecture uses batch normalization and the ReLU activation function in the first two fully connected layers and the linear activation function in the last fully connected layer. The regressor was trained on bucket feature of stego images, where a bucket of 6 detectors trained for payload 0.1, 0.2, 0.3, 0.4, 0.5, 0.6 bpp.

C. ResNet-50

Deep Residual Network based quantitative steganalysis built on [13] is adapted from pretrained ResNet-50. ResNet-50 [14] is divided into 5 stages, as shown in Fig. 3, where stage 1 consists of the initial convolution layer and max pooling. The image input (e.g., $224 \times 224 \times 3$) passes through the initial convolutional layer with a 7×7 kernel, stride 2, and 64 filters, followed by Batch Normalization (BN) and ReLU activation. Then, a 3×3 max pooling layer with stride 2. Stage 2 and subsequent stages are bottleneck blocks. ResNet50 uses bottleneck blocks for computational efficiency. Each bottleneck block typically consists of three convolutional layers and an identity block. In each convolutional sub-layer, it is followed by Batch Normalization and ReLU (except after the final 1×1 layer within the block, where ReLU is applied after the shortcut addition). After all residual block stages are passed, there is an Average Pooling layer to reduce the spatial dimensions of the feature map to 1×1 . The output then enters the Fully Connected (FC) layer for regression [13].

D. VGG-16

VGG-16 [15] is a CNN commonly used for digital image classification, and in [13] VGG-16 was used for quantitative steganalysis. ResNet-50 built on [13] compared its performance with VGG-16 for quantitative steganalysis. The VGG-16 architecture [15] as shown in Fig. 4 uses several convolution blocks in sequence. Each block has a different number of filters and filter sizes, enabling the network to learn increasingly complex features. After several convolutional layers, there is

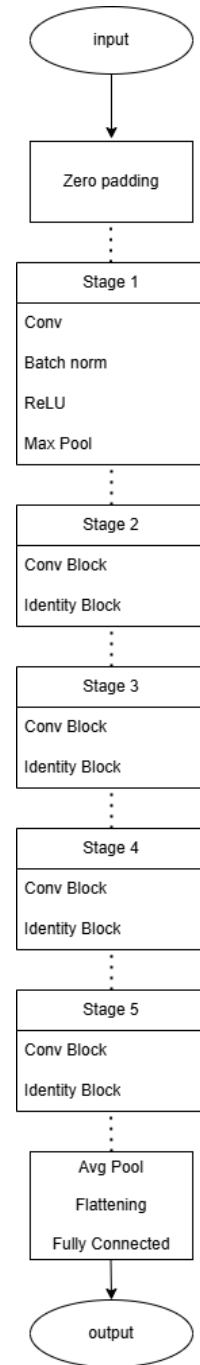


Fig. 3. ResNet-50 [14]

a pooling layer that reduces the data dimensions. This helps reduce overfitting and accelerates the learning process. At the end of the network, three fully connected layers are used for regression.

III. RESEARCH METHOD

The dataset used in this study was obtained from Bossbase 1.01 [18], which has been used in BOSS competition (Break Our Steganographic System). It consist of 10,000 images in PGM format, each measuring 512×512 pixels. Bossbase 1.01

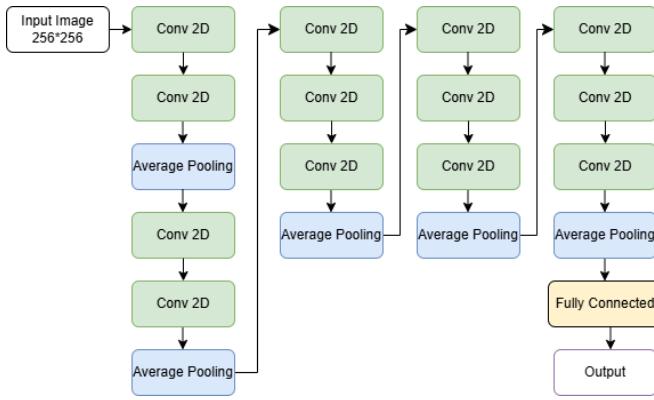


Fig. 4. VGG-16 [15]

has been used widely in steganalysis research [16], [19]–[22]. An example of data in Bossbase 1.01 is shown in Fig. 5. This dataset was chosen because of its sufficient size and diversity that make it a common benchmark in steganalysis research, which enables other studies to reproduce the results. The dataset size of 10,000 images is considered large enough to train and test the model without overfitting, and the images are diverse, having been captured using seven different digital cameras. With seven cameras, the model is forced to learn to recognize subtle patterns resulting from the steganography process, rather than patterns originating from the cameras themselves. This diversity helps the steganalysis model learn to detect anomalies from the message insertion process, rather than from the unique characteristics of a single camera type. The Bossbase 1.01 dataset consists of cover images (without secret messages). Secret messages are embedded in the cover images using a steganography algorithm. The steganography algorithm used in this study is S-Uniward (Spatial-Universal Wavelet Relative Distortion) with a message length between 0.1 and 0.6 bpp (bits per pixel). A stego data is not yet publicly available, so the authors created a new dataset, with the following steps:

- The size of all images was changed from 512*512 to 256*256
- Embedding was performed using the S-Uniward algorithm with message length between 0.1 and 0.6 bpp using MATLAB.
- The data was divided into training, testing, and validation data as in Table I.

There were 24000 training data, 6000 validation data, and 30000 testing data in range 0.1 until 0.6 bpp. The proportion of testing data is larger than that of training data to emphasize evaluation and validation, ensuring that the model can serve as a reference for quantitative steganalysis. Data division with these proportions has also been used in previous research [16], namely the GBRAS Net classifier for binary steganalysis. The research procedure consists of data preprocessing, model training and performance evaluation as follows.



Fig. 5. Sample of PGM Image in Bossbase 1.01 [18]

TABLE I
DATA TRAINING, VALIDATION AND TESTING

	Total
Train Data	24000
Validation Data	6000
Test Data	30000

A. Data Preprocessing

In the data preprocessing stage, input image are processed using an SRM Filter [20] as shown in Fig. 6. The numbers in each box on the SRM filter are the weights or coefficients of a convolution filter. Each box is designed as a high-pass filter (HPF). The HPF function suppresses image content (such as objects or scenery) and amplifies noise or subtle artifacts that are invisible to the human eye. Secret messages embedded into images using the S-Uniward algorithm will leave traces in the form of very small statistical changes. These filters are highly sensitive to such traces. The sum of all numbers in most SRM filters is zero (for example, the filter in row 1, which when added up is 0), ensuring that when the filter passes over a flat area of the image (where the colors are uniform), the result will be close to zero. This effectively removes unimportant parts of the image and leaves only those parts that have drastic changes, such as edges. Using these 30 filters yields 30 residual maps, which are then analyzed by the CNN model for steganalysis.

B. Model Training

The feature extraction in this study was conducted using the GBRAS Net. GBRAS Net is one of the binary steganalysis models with superior accuracy, so it is proposed for the feature extraction layer for the deep learning regressor. The GBRAS network [16] has a preprocessing block using 30 SRM Filter in the first convolution. The GBRAS network has a convolution module for feature extraction with depthwise separable convolution layers and 2 skip connections. In the GBRAS Net convolution module, the average pooling layer after batch normalization aims to reduce the spatial dimensions of the data. Global Average Pooling (GAP) is used before the classification layer.

Regression is performed by replacing the softmax activation function with a linear activation function in the output layer,

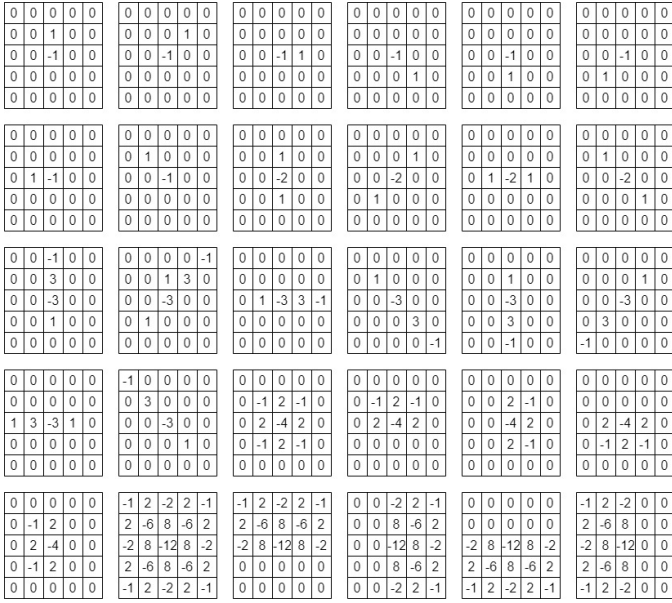


Fig. 6. SRM Filter [20]

and changing the loss function from categorical crossentropy to MSE. The output from the CNN will directly become the predicted value, without exponential transformation as performed by Softmax. A fully connected layer is used for regression. The deep learning regressor based on GBRAS Net is referred to as the GBRAS Net regressor. The GBRAS Net regressor architecture is shown in Fig. 7. The deep learning parameters used to perform the learning process are learning rate: 0.001, batch size of 32, epoch of 100 and Adam as the optimizer algorithm. The implementation of CNN for quantitative steganalysis in this study uses Python 3.8.5 and Tensorflow 2.5.0 on Google Colaboratory using a Tesla T4 GPU (16GB).

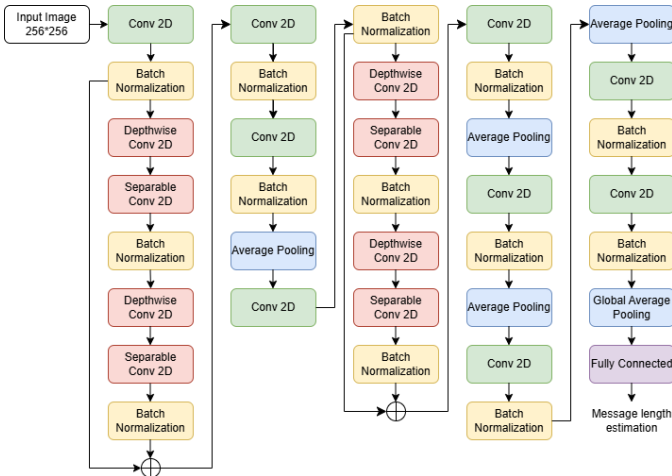


Fig. 7. Architecture of GBRAS Net Regressor

C. Performance Evaluation

The model was tested using the data from Bossbase 1.01. The tests used were Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE). The goal of regression is to predict continuous values, and these metrics measure how well the model's predictions compare to the actual values. MSE measures the average of the squared differences between the predicted values and the actual values. By squaring the differences, MSE gives greater weight to large errors. RMSE is the square root of MSE. This makes RMSE easier to interpret than MSE. MAE measures the average of the absolute differences between the predicted values and the actual values. Unlike MSE and RMSE, MAE treats all errors proportionally, without giving greater weight to large errors.

IV. RESULTS AND ANALYSIS

The testing was conducted by comparing the performance of the proposed model with other quantitative steganalysis studies. The test scenario is to compare the performance of the GBRAS Net Regressor with Resnet-50 [13], VGG-16 [15], and Ye Net Regressor [3]. Table II shows the comparison of MSE, RMSE, and MAE values for these models. The proposed model demonstrates better performance than other deep learning models for message length estimation.

TABLE II
COMPARISON WITH ANOTHER QUANTITATIVE STEGANALYSIS RESEARCH

No.	Model	MSE	RMSE	MAE
1	ResNet-50	0.0294	0.1714	0.1500
2	VGG-16	0.0297	0.1722	0.1499
3	Ye Net regressor	0.0183	0.1353	0.1085
4	proposed model	0.0182	0.1349	0.1064

ResNet-50 showed poor performance (MSE 0.0294, RMSE 0.1714, MAE 0.1500) and VGG-16 model has very high MSE (0.0297), RMSE (0.1722), and MAE (0.1499) values, indicating poor prediction accuracy. The Ye Net regressor shows good performance with MSE (0.0183), RMSE (0.1353), and MAE (0.1085). The proposed model shows the lowest MSE (0.0182), RMSE (0.1349), and MAE (0.1064) values among all tested models. Based on RMSE, there is a significant improvement compared to ResNet-50 (0.0365) and VGG-16 (0.0373). This confirms that the proposed model is also better at reducing the mean squared error compared to the ResNet-50 and VGG-16 models. The improvement is very small compared to the Ye Net regressor (0.0004). This confirms that the Ye Net regressor is already quite good at minimizing the root mean squared error, and the proposed model can only provide a marginal improvement in this metric. Based on the MAE metric, the proposed method reduces the MAE of ResNet 50 by 0.0436, VGG-16 by 0.0435, Ye Net regressor by 0.0021. Although the difference is small, it still demonstrates that the proposed model is slightly more effective in reducing the mean absolute error even compared to an already good model. A significant improvement compared to ResNet-50 (0.0436)

and VGG-16 (0.0435). This indicates that the proposed model substantially reduces the mean absolute error compared to the aforementioned architectures. The achieved MSE of 0.0182 is lower than ResNet 50 by [13] (0.0294) . It also lower than VGG-16 by [13](0.0297) and Ye Net regressor (0.0183) as in [3]. The RMSE and MAE values also show a decrease compared to ResNet-50, VGG-16, and Ye Net regressor.

V. CONCLUSION AND FUTURE WORKS

This study aims to develop more effective predictive model for secret message length estimation. Estimation of the length of secret messages in previous works shows high MAE values. So, this study proposed GBRAS Net regressor for estimation of secret message length. From the comparison between the proposed model and several other deep learning regressors, it can be seen that the proposed model has significantly better performance (lower error values) compared to the other three models (Resnet 50, VGG 16, and Ye Net regressor). It was because quantitative steganalyzer performance depends on feature extraction, so improvements to the CNN architecture will also improve the performance of the quantitative steganalyzer.

Future work may explore steganalysis for secret message extraction by using deep learning. Research on secret message extraction by using deep learning has not been conducted. Therefore, successfully developing such a system would constitute a significant contribution to the field of steganalysis.

ACKNOWLEDGMENT

The authors would like to thank the Bandung Institute of Technology for providing access to the software licenses and computing facilities used in this research.

REFERENCES

- [1] M. Chaumont, *Deep learning in steganography and steganalysis*. 2020.
- [2] R. Munir, "Kriptografi," 2019.
- [3] M. Chen, M. Boroumand, and J. Fridrich, "Deep learning regressors for quantitative steganalysis," in *IS and T International Symposium on Electronic Imaging Science and Technology*, 2018.
- [4] J. Kodovský and J. Fridrich, "Quantitative steganalysis using rich models," in *Media Watermarking, Security, and Forensics 2013*, vol. 8665, 2013.
- [5] Y. Sun and T. Li, "A method for quantitative steganalysis based on deep learning," in *2019 2nd International Conference on Information Systems and Computer Aided Education, ICISCAE 2019*, 2019.
- [6] J. Kodovský, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," in *IEEE Transactions on Information Forensics and Security*, vol. 7, 2012.
- [7] K. Liu, J. Yang, and X. Kang, "Ensemble of cnn and rich model for steganalysis," in *International Conference on Systems, Signals, and Image Processing*, 2017.
- [8] M. H. Menori and R. Munir, "Blind steganalysis for digital images using support vector machine method," in *2016 International Symposium on Electronics and Smart Devices, ISESD 2016*, 2017.
- [9] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, 2012.
- [10] A. Selvaraj, A. Ezhilarasan, S. L. J. Wellington, and A. R. Sam, "Digital image steganalysis: A survey on paradigm shift from machine learning to deep learning based techniques," *IET Image Processing*, vol. 15, pp. 504–522, 2 2021.
- [11] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, 2017.
- [12] A. Singhal and P. Bedi, "Blind quantitative steganalysis using cnn–long short-term memory architecture," in *Strategic System Assurance and Business Analytics*, pp. 175–186, Singapore: Springer Singapore, 2020.
- [13] A. Singhal and P. Bedi, *Universal Quantitative Steganalysis Using Deep Residual Networks*. 2022.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, 2016.
- [15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [16] T. S. Reinel, A. A. H. Brayan, B. O. M. Alejandro, M. R. Alejandro, A. G. Daniel, A. G. J. Alejandro, B. J. A. Buenaventura, O. A. Simon, I. Gustavo, and R. P. Raul, "Gbras-net: A convolutional neural network architecture for spatial image steganalysis," *IEEE Access*, vol. 9, 2021.
- [17] R. K. Dewi and R. Munir, "Perbandingan berbagai metode steganografi pada citra digital," *Jurnal Informatika Polinema*, vol. 9, no. 3, 2023.
- [18] P. Bas, T. Filler, and T. Pevný, "'Break our steganographic system': The ins and outs of organizing BOSS," in *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6958 LNCS, p. ??, Springer Berlin Heidelberg, 2011.
- [19] G. Xu, H. Wu, and Y. Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Processing Letters*, vol. 23, no. 5, pp. 708–712, 2016.
- [20] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 12, 2017.
- [21] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2018.
- [22] M. Yedroudj, F. Comby, and M. Chaumont, "Yedroudj-net: An efficient cnn for spatial steganalysis," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 2092–2096, IEEE, 2018.