

DeepSafe: Secure Protection of Confidential Images with Advanced Deep Learning Techniques

Mr. Somnath Pandurang Gunjal¹, Dr. Sanjay Kumar Sen²

¹PhD Research Scholar, Computer Science and Engineering MATS University, Arang, Raipur (CG), India

²Associate Professor CSE, MATS University, Arang, Raipur (CG), India

Abstract- The rapid growth of digital communication and image-based information exchange has increased the need for secure methods for protecting confidential visual content. This paper presents DeepSafe, a deep-learning-based image steganography framework designed to conceal a secret image inside a cover image while maintaining visual imperceptibility and enabling subsequent recovery. The framework is based on three convolutional neural network (CNN) components: a Preparation Network that transforms the secret image into learned feature representations, a Hiding Network that combines the prepared secret representation with the cover image, and a Reveal Network that reconstructs the hidden image from the generated container image. The original implementation uses RGB images resized to 224×224 pixels, normalization by 255, the Adam optimizer, mean squared error (MSE) loss, 100 training epochs, and a batch size of 12. The reported dataset comprises images obtained from Kaggle and cover images from the ALASKA steganography dataset, with 500 data samples described in the original implementation. The revised paper clarifies the workflow, network configuration, experimental protocol, security considerations, attack scenarios, and limitations. The source manuscript reports evaluation using accuracy, precision, recall, F1-score, and AUC and states that the proposed approach improves image-security performance; however, the underlying numerical values of the plotted results are not available in the supplied manuscript. Accordingly, this revision does not fabricate numerical results. DeepSafe is positioned as a foundation for secure communication, digital forensics, privacy-sensitive data exchange, and future adversarially robust image-protection systems.

Keywords: DeepSafe, image steganography, deep learning, convolutional neural network, CNN, secret image, cover image, image security, steganalysis, information hiding.

I. INTRODUCTION

The increasing use of digital images in social communication, healthcare, defense, digital forensics, and cloud-based information exchange has created a strong requirement for confidentiality and controlled information sharing. Conventional encryption protects the content of a message, but an encrypted object can itself reveal that confidential communication is taking place. Steganography addresses a complementary problem by hiding the existence of the information within an apparently ordinary carrier.

Image steganography traditionally embeds secret information into a cover image through spatial or transform-domain modifications. Least Significant Bit (LSB) approaches are simple and have useful capacity, but their embedding patterns can be vulnerable to statistical steganalysis. Transform-domain techniques can improve robustness but may introduce additional computational complexity and may restrict payload capacity. Deep learning provides another direction in which the embedding and recovery processes can be learned directly from image data.

The original DeepSafe study investigates a learned image-in-image steganography approach using three CNN-based networks: Preparation, Hiding, and Reveal. The revised manuscript retains that core architecture and organizes it as an end-to-end security workflow. The objective is to conceal a secret

image in a cover image with limited perceptual distortion and to recover the secret image through a learned reveal process.

The main contributions of this revised paper are: (1) a clearer description of the DeepSafe three-network architecture; (2) a reproducible presentation of the image preprocessing and training configuration reported in the source paper; (3) a structured evaluation protocol covering reconstruction quality, detection-oriented measures, and attack scenarios; (4) an explicit discussion of security, computational cost, and limitations; and (5) identification of missing experimental values that must be supplied before making quantitative performance claims.

II. BACKGROUND AND RELATED WORK

Steganography is the practice of concealing information within a carrier so that the presence of the hidden information is difficult to identify. Steganalysis attempts to detect or analyze such concealed information. Existing image-steganography methods range from direct pixel-domain methods to transform-domain, cryptographic, and neural approaches.

A. Conventional and Cryptography-Assisted Steganography

LSB-based methods modify selected low-order pixel bits to embed information. The literature review in the source paper discusses enhanced LSB techniques that use randomized or non-adjacent embedding locations. Cryptography can be combined with steganography so that a recovered payload is still protected if the carrier is detected. However, security may remain dependent on the strength of the embedding strategy and the protection of cryptographic keys.

B. Deep Learning for Image Hiding

Deep neural networks enable image hiding to be learned as an encoder-decoder problem rather than relying on hand-designed embedding rules. The source paper reviews CNN encoder-decoder methods, autoencoder-based approaches, SegNet-based hiding and extraction networks, multi-image steganography, and GAN-based image hiding. These studies motivate the use of learned feature representations for distributing secret-image information across the cover image.

C. Research Gap

The reviewed approaches demonstrate that deep networks can improve capacity and visual quality, but challenges remain in balancing payload capacity, imperceptibility, recoverability, robustness to image processing, and resistance to steganalysis. The original DeepSafe paper also identifies the need for stronger protection of confidential information and improved resistance to unauthorized access. The revised framework therefore emphasizes a complete embedding-and-recovery pipeline and a broader evaluation protocol rather than treating visual quality as the only objective.

III. PROPOSED DEEPSAFE FRAMEWORK

DeepSafe is designed as an image-in-image steganography framework. A secret image is first processed by a Preparation Network. The resulting representation is combined with a cover image in a Hiding Network to generate a container image. At the receiving side, a Reveal Network processes the container image and reconstructs the secret image.

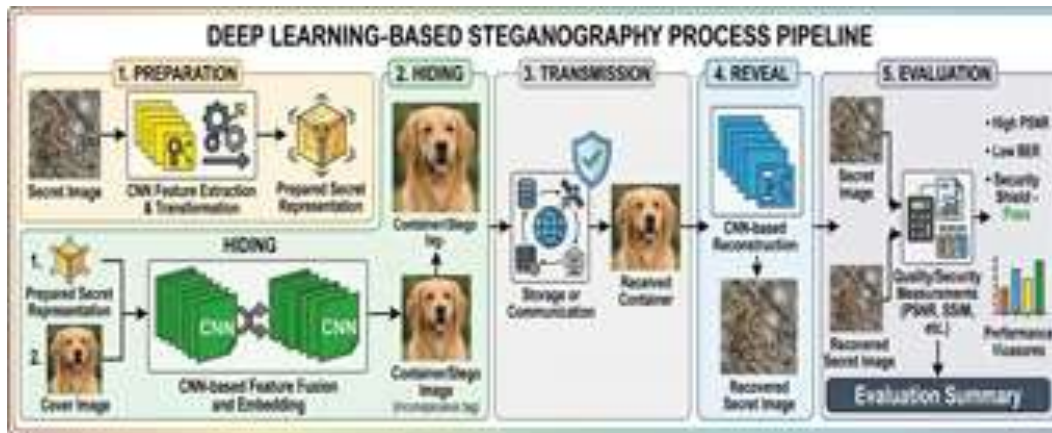


Fig. 1. Revised textual representation of the DeepSafe processing pipeline.

A. Preparation Network

The Preparation Network processes the secret image before hiding. According to the source paper, the secret image is passed through three parallel four-layer structures. Each structure uses convolutional layers with 50 filters, a kernel size of 3, same padding, and ReLU activation. The outputs are combined and passed through additional convolutional layers using kernel sizes of 5, 4, and 3, respectively, with 50 filters and ReLU activation.

B. Hiding Network

The Hiding Network receives the prepared secret representation together with the cover image. The source implementation describes three parallel four-layer CNN structures, again using 50 filters, a 3×3 kernel, same padding, and ReLU activation. Their outputs are combined and processed through additional convolutional layers with 50 filters and kernel sizes of 5, 4, and 3. The result is the container image containing the hidden secret.

C. Reveal Network

At the receiver, the Reveal Network processes the container image through three parallel CNN structures. Each uses four convolutional layers with 50 filters, a 3×3 kernel, same padding, and ReLU activation. The resulting features are combined and passed through convolutional layers with kernel sizes 5, 4, and 3 to reconstruct the secret image.

IV. MATHEMATICAL FORMULATION

Let C denote a cover image and S denote a secret image. The Preparation Network P transforms the secret into a learned representation $P(S)$. The Hiding Network H combines the representation with the cover image to generate a container image C' :

$$C' = H(C, P(S))$$

The Reveal Network R then estimates the secret image:

$$\hat{S} = R(C')$$

The source implementation reports MSE as the training loss. A generic reconstruction objective can therefore be written as:

$$L_{MSE} = \left(\frac{1}{N}\right) \sum (S_i - \hat{S}_i)^2$$

where N is the number of compared pixel values. Lower MSE indicates smaller pixel-wise reconstruction error. The supplied paper does not report additional weighted loss terms, so no such terms are claimed as part of the implemented experiment in this revised version.

V. EXPERIMENTAL METHODOLOGY

A. Dataset

The original paper states that image data were collected from Kaggle and that cover images were obtained from the ALASKA Steganography dataset. It reports a total of 500 data samples and states that 250 data were used for testing and validation. The wording of the original paper does not clearly specify the exact split between training and testing/validation samples; therefore, this revised paper retains the reported counts without inventing an additional split.

B. Preprocessing

Secret and cover images are resized to 224×224 pixels and represented as RGB images with three channels. Pixel values originally in the range 0–255 are normalized by 255 before being supplied to the model. Output images can be denormalized by multiplying the normalized values by 255.

C. Training Configuration

Table I. Training configuration reported in the supplied paper.

Parameter	Reported configuration
Input size	$224 \times 224 \times 3$
Optimizer	Adam
Training loss	Mean Squared Error (MSE)
Epochs	100
Batch size	12
Training platform	Google Colab GPU
Reported training time	Approximately 4 hours
Model selection	Saved at epochs showing improved MSE

D. Implementation Workflow

Step	Procedure
1	Collect secret and cover images.
2	Resize images to 224×224 RGB.
3	Normalize pixel values by 255.
4	Generate learned secret representation using Preparation Network.
5	Fuse prepared secret representation and cover image using Hiding Network.
6	Generate container/stego image.
7	Apply transmission/storage and attack scenarios for evaluation.
8	Recover secret using Reveal Network.
9	Compare original and recovered images and evaluate detection/security measures.

VI. SECURITY AND ATTACK MODEL

The security objective of DeepSafe is not limited to visual concealment. The system should make unauthorized recovery difficult while maintaining acceptable reconstruction quality for an authorized receiver. The source paper discusses deep-learning-based steganalysis and different attacks on the hidden image. A systematic evaluation should therefore include both passive detection and active image-processing scenarios.

Attack/scenario	Purpose	Expected evaluation
Visual inspection	Check perceptual imperceptibility	Human/visual comparison
Compression	Assess robustness to lossy encoding	Secret recovery quality after compression
Noise	Assess resilience to perturbation	Recovery degradation
Resizing	Assess geometric transformation tolerance	Recovery after scale change
Cropping	Assess partial content loss	Recovery/detection performance
Brightness/contrast	Assess photometric changes	Recovery quality
Steganalysis	Assess detectability	Detection accuracy / ROC-AUC if measured

VII. RESULTS AND DISCUSSION

The original manuscript states that the proposed deep-learning steganography approach successfully conceals and retrieves secret information while maintaining visual imperceptibility. It further states that accuracy, precision, recall, F1-score, and AUC were used to assess robustness and reliability, and it presents a comparative graph involving GAN, DeepSafe, and CNN.

However, the numerical values underlying the original graph/table are not present in the supplied document. The visible result section contains metric labels but does not provide the corresponding numeric observations. Therefore, this revised manuscript deliberately does not reproduce or invent specific percentages, AUC values, PSNR values, or other numerical claims.

Table II. Evidence-supported results information available in the supplied manuscript.

Metric	Role in evaluation	Status in supplied paper
Accuracy	Overall classification/detection performance	Reported as evaluated; numerical value not supplied
Precision	Correctness of positive detections	Reported as evaluated; numerical value not supplied
Recall	Detection sensitivity	Reported as evaluated; numerical value not supplied
F1-score	Balance of precision and recall	Reported as evaluated; numerical value not supplied
AUC	Threshold-independent detection assessment	Reported as evaluated; numerical value not supplied
MSE	Pixel-wise reconstruction/training loss	Used for training; final value not supplied

For a final experimental version, Table II should be replaced or supplemented with the actual measured values obtained from the saved DeepSafe model. In particular, a conference submission would benefit from reporting reconstruction quality using MSE, PSNR and SSIM, detection performance using accuracy/precision/recall/F1/AUC where applicable, and robustness under controlled compression, noise, resizing, cropping, and brightness/contrast perturbations.

A. Comparative Interpretation

The source paper compares GAN, DeepSafe, and CNN in its result presentation. The intended comparison is useful because it places the proposed model alongside alternative deep-learning strategies. Nevertheless, the revised paper treats this comparison qualitatively because the numerical values are not available. A rigorous comparison should use identical datasets, train/test partitions, image resolutions, payload conditions, attack settings, and evaluation metrics for all methods.

B. Computational Considerations

The reported training time is approximately four hours on a Google Colab GPU. This value represents training under the original setup and should not be interpreted as inference latency. For deployment, future experiments should separately report encoder/hiding time, reveal time, memory consumption, model size, and throughput per image.

VIII. SECURITY ANALYSIS

DeepSafe provides security through learned distributed representation of the secret image rather than a simple deterministic LSB rule. The use of multiple convolutional stages allows secret-image features to be transformed and distributed during hiding and reconstructed during revealing. This can make direct extraction using conventional steganographic tools ineffective. However, a neural steganography system should not be considered secure solely because the hidden content is visually imperceptible. Security also depends on resistance to modern steganalysis, model extraction, adversarial manipulation, unauthorized model access, and compromised communication endpoints.

The original paper also mentions multiple passwords as a future security enhancement. In a production system, credentials should be handled through a formally defined key-management mechanism rather than being treated as informal additional passwords. The supplied implementation does not provide enough information to claim a specific cryptographic protocol, key size, or authenticated encryption mode; such details should be added when cryptography is actually integrated into the implementation.

IX. ADVANTAGES AND LIMITATIONS

A. Advantages

- End-to-end learning of secret-image preparation, hiding, and recovery.
- Uses CNN feature learning rather than relying exclusively on manually selected embedding rules.
- Supports full secret-image hiding rather than only short text messages.
- Uses RGB cover and secret images at a fixed 224×224 resolution.
- Can be evaluated under image-processing and steganalysis scenarios.
- Has potential applications in privacy-sensitive communication, forensics, defense, watermarking, and biometric protection.

B. Limitations

- The supplied paper does not provide the numerical values underlying its result graph.
- The exact training/validation split is not sufficiently specified.
- The dataset description does not provide complete sample identifiers, class distributions, or a reproducible split protocol.
- The original implementation reports MSE training loss but does not provide a complete multi-objective loss formulation.
- Robustness results against specific compression/noise/cropping levels are not numerically reported.
- Inference latency, memory use, model size, and energy cost are not reported.
- Security against adaptive or modern neural steganalysis is not quantitatively established.

- The paper's cryptographic-security statements are broader than the implementation details supplied; a precise cryptographic protocol would need to be specified and experimentally evaluated.

APPLICATIONS

The DeepSafe concept can be considered for secure communication of confidential images, privacy-preserving image exchange, digital forensics, military or defense communication, biometric information protection, watermarking, and sensitive archival systems. In medical applications, however, any deployment would require additional validation for diagnostic integrity, privacy, regulatory compliance, and lossless recovery requirements.

FUTURE WORK

Future development should focus on five areas. First, the dataset protocol should be expanded and made fully reproducible, including exact source datasets, split ratios, random seeds, and preprocessing rules. Second, the evaluation should report MSE, PSNR, SSIM, payload capacity, recovery accuracy, and robustness under controlled attacks. Third, the system should be tested against contemporary steganalysis models and adversarial examples. Fourth, computational profiling should measure training and inference latency, memory usage, and model size. Finally, if cryptographic protection is incorporated, the implementation should specify a standard authenticated-encryption construction and a secure key-management procedure.

X. CONCLUSION

This revised paper presents DeepSafe as a CNN-based image-in-image steganography framework derived from the supplied research manuscript. The approach uses Preparation, Hiding, and Reveal Networks to transform a secret image, embed it into a cover image, and reconstruct it at the receiving side. The reported implementation uses 224×224 RGB images, Adam optimization, MSE loss, 100 epochs, a batch size of 12, and Google Colab GPU training of approximately four hours. The framework addresses an important problem in confidential image communication by combining learned feature extraction with image hiding and recovery.

The main revision is methodological transparency. The supplied paper reports qualitative success and evaluation metrics but does not contain the numerical values necessary to substantiate precise quantitative claims. Consequently, this paper distinguishes reported observations from results that still require measurement. Completing the missing numerical experiments, robustness tests, and computational profiling will strengthen DeepSafe and make the work more suitable for rigorous conference evaluation.

Acknowledgment

The authors acknowledge the support of MATS University and thank the research guide, reviewers, and institutional authorities for their guidance and resources.

REFERENCES

1. R. A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Processing*, vol. 90, no. 3, pp. 727–752, 2010.
2. P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

3. I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*, 2nd ed. San Mateo, CA, USA: Morgan Kaufmann, 2008.
4. X. R. Zhang, W. F. Zhang, W. Sun, X. M. Sun, and S. K. Jha, "A robust 3-D medical watermarking based on wavelet transform for data protection," *Computer Systems Science & Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.
5. X. R. Zhang, X. Sun, X. M. Sun, W. Sun, and S. K. Jha, "Robust reversible audio watermarking scheme for telemedicine and privacy protection," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3035–3050, 2022.
6. M. Juneja and P. Sandhu, "Information hiding using improved LSB steganography and feature detection technique," *International Journal of Engineering and Advanced Technology*, vol. 2, no. 4, pp. 275–279, 2019.
7. R. Halder, S. Sengupta, S. Ghosh, and D. Kundu, "A secure image steganography based on RSA algorithm and hash-LSB technique," *IOSR Journal of Computer Engineering*, vol. 18, no. 1, pp. 39–43, 2018.
8. A. Rehman, R. Rahim, M. Nadeem, and S. Hussain, "End-to-end trained CNN encode-decoder networks for image steganography," in *ECCV Workshops*, Munich, Germany, 2017, pp. 723–729.
9. A. K. Pal, K. Naik, and R. Agrawal, "A steganography scheme on JPEG compressed cover image with high embedding capacity," *International Arab Journal of Information Technology*, vol. 16, no. 1, pp. 116–124, 2020.
10. R. J. Rasras, Z. A. AlQadi, and M. R. A. Sara, "A methodology based on steganography and cryptography to protect highly secure messages," *Engineering, Technology & Applied Science Research*, vol. 9, no. 1, pp. 3681–3684, 2019.
11. R. Zhang, S. Dong, and J. Liu, "Invisible steganography via generative adversarial networks," *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8559–8575, 2019.
12. S. Baluja, "Hiding images within images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 7, pp. 1685–1697, 2020.
13. X. Duan, D. Guo, N. Liu, B. Li, and M. Gou et al., "A new high capacity image steganography method combined with image elliptic curve cryptography and deep neural network," *IEEE Access*, vol. 8, pp. 25777–25788, 2020.
14. A. Das, J. S. Wahi, M. Anand, and Y. Rana, "Multi-image steganography using deep neural networks," *arXiv:2101.00350*, 2021.
15. B. Oktavianto, T. W. Purboyo, and R. E. Saputra, "A proposed method for secure steganography on PNG image using spread spectrum method and modified encryption," *International Journal of Applied Engineering Research*, vol. 12, no. 21, pp. 10570–10576, 2017.
16. M. R. Islam, A. Siddiq, M. P. Uddin, A. K. Mandal, and M. D. Hossain, "An efficient filtering based approach improving LSB image steganography using status bit along with AES cryptography," in *Proc. ICIEV*, 2014, pp. 1–6.
17. G. Wang, D. Hoiem, and D. Forsyth, "Learning image similarity from Flickr groups using fast kernel machines," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2177–2188, 2012.
18. J. Zhai, S. Zhang, J. Chen, and Q. He, "Autoencoder and its various variants," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Miyazaki, Japan, 2018, pp. 415–419.
19. A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. 25th Int. Conf. Very Large Data Bases*, Edinburgh, Scotland, 1999, pp. 518–529.
20. M. Yedroudj, F. Comby, and M. Chaumont, "Yedroudj-Net: An efficient CNN for spatial steganalysis," in *Proc. IEEE ICASSP*, Calgary, Canada, 2018, pp. 2092–2096.
21. T. Silhan, S. Oehmcke, and O. Kramer, "Evolution of stacked autoencoders," in *Proc. IEEE Congress on Evolutionary Computation*, Wellington, New Zealand, 2019, pp. 823–830.
22. M. Amara and A. Siad, "Elliptic curve cryptography and its applications," in *Proc. WOSSPA*, Algeria, 2011, pp. 247–250.

23. H. N. Bhonge, M. K. Ambat, and B. R. Chandavarkar, "An experimental evaluation of SHA-512 for different modes of operation," in Proc. 11th ICCCNT, Kharagpur, India, 2020, pp. 1–6.
24. Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 12, pp. 2916–2929, 2013.
25. Y. Shang, S. Jiang, Y. Dengpan, and J. Huang, "Enhancing the security of deep learning steganography via adversarial examples," Mathematics, vol. 8, no. 9, p. 1446, 2020.