AI & CYBER FORUM
AN INTERNATIONAL JOURNAL
OPEN ACCESS
ISSN 3117-6399
Original Research Article
Check for updates

# Machine Learning-Assisted Image Steganography Using Intelligent Pixel Classification and Dynamic LSB Data Embedding

**Pankaj Nandan**

*Department of Computer Applications, Government Degree College, Ramkote, Jammu and Kashmir 184205, India*

Corresponding author: **Pankaj Nandan** | E-mail: **prof.pankajnandan@gmail.com**

**Citation:** Pankaj Nandan (2024). Machine Learning-Assisted Image Steganography Using Intelligent Pixel Classification and Dynamic LSB Data Embedding. *AI & Cyber Forum: An International Journal.* **DOI: https://doi.org/10.51470/AI.2024.3.1.04**

Received 07 January 2024 | Revised 04 February 2024 | Accepted 06 March 2024 | Available Online April 08, 2024

## *Abstract*

*The increasing demand for secure digital communication has encouraged the development of intelligent image steganography techniques capable of concealing confidential information while maintaining the visual quality of digital images. Conventional Least Significant Bit (LSB) substitution provides a simple and computationally efficient embedding mechanism; however, uniform pixel selection may introduce unnecessary modifications in smooth image regions and increase the statistical detectability of embedded information. This study proposes a machine learning-assisted image steganography framework that integrates intelligent pixel classification with dynamic LSB data embedding to improve the selection of suitable embedding locations. The proposed approach employs image-derived features, including local variance, gradient magnitude, texture complexity, local entropy, and neighbouring-pixel differences, to classify pixels according to their embedding suitability. A supervised machine learning classifier, such as a Random Forest model, is proposed to estimate the suitability of individual pixels or local image regions for information hiding. Based on the classification output, a dynamic embedding mechanism determines whether a pixel should be excluded from embedding or assigned an appropriate embedding rate. Secret information is subsequently embedded through an adaptive LSB substitution procedure, while a synchronized extraction mechanism enables recovery of the hidden payload. The proposed framework is designed to balance embedding capacity, visual imperceptibility, computational efficiency, and statistical detectability. Its performance is intended to be evaluated using standard image-quality metrics, including Mean Squared Error, Peak Signal-to-Noise Ratio, Structural Similarity Index Measure, embedding rate, and machine-learning-based steganalysis. The study establishes a reproducible methodological framework for investigating the application of supervised learning to adaptive spatial-domain image steganography and provides a basis for future development of intelligent information-hiding systems.*

***Keywords:*** *Image steganography; machine learning; adaptive LSB substitution; pixel classification; dynamic data embedding.*

## 1. Introduction

The rapid development of digital communication technologies has increased the exchange of multimedia information through online platforms, cloud environments, institutional networks, and mobile communication systems. Alongside these developments, protecting confidential information against unauthorized access, interception, and disclosure has become an important research priority. Cryptographic techniques protect information by converting readable messages into encrypted representations, whereas steganography attempts to conceal the existence of confidential communication by embedding secret information within ordinary digital media. Among the different forms of steganography, image-based information hiding has received considerable attention because digital images provide large numbers of pixels that can potentially accommodate secret information without producing obvious visual changes.

Least Significant Bit substitution is one of the most widely investigated spatial-domain image steganography techniques because of its computational simplicity and relatively high theoretical embedding capacity. The conventional approach replaces the least significant bit of selected pixel values with secret-message bits, generally producing only minor changes in pixel intensity [1]. However, conventional LSB methods frequently employ sequential or uniform embedding strategies without adequately considering the local characteristics of the cover image. Consequently, smooth regions, textured areas, and edge-containing regions may undergo similar embedding operations despite exhibiting different levels of perceptual sensitivity. Such uniform modification can introduce unnecessary distortion and may create statistical irregularities that can be identified through steganalysis. Adaptive image steganography attempts to address these limitations by selecting embedding locations according to image characteristics.

© ***Authors:*** *Published in AI & Cyber Forum: An International Journal under the CC BY-NC-ND 4.0 license (http://creativecommons.org/licenses/by-nc-nd/4.0/). No commercial use or modifications permitted.*

Edge strength, local variance, entropy, texture complexity, and neighbouring-pixel relationships are among the features that can be used to identify regions with different embedding suitability. Nevertheless, conventional threshold-based selection methods may be sensitive to image content and may not adequately represent the complex relationships between multiple image features. Machine learning offers an alternative approach by learning patterns from image-derived features and using the resulting classification model to guide embedding decisions. Machine learning-assisted steganography can potentially improve pixel selection by identifying image regions that are more suitable for information hiding under defined embedding conditions. Supervised classification algorithms can analyse multiple characteristics simultaneously and estimate the suitability of individual pixels or local regions. The classification output can then guide a dynamic embedding mechanism that adjusts the embedding rate according to the predicted characteristics of the cover image. In this context, the integration of intelligent pixel classification with modified LSB substitution provides an opportunity to develop more flexible embedding systems than conventional uniform LSB approaches [2]. The present study proposes a machine learning-assisted image steganography framework that combines local image-feature extraction, supervised pixel classification, and dynamic LSB data embedding. The primary objective is to investigate how intelligent pixel selection can guide the allocation of secret information while maintaining acceptable image quality and embedding efficiency. The proposed framework also establishes an experimental evaluation strategy for comparing the intelligent embedding method with conventional LSB substitution and other adaptive approaches.

## 2. Research Objectives and Proposed Contributions

The principal objective of this research is to develop a machine learning-assisted image steganography framework that improves embedding-location selection through intelligent pixel classification and dynamic LSB substitution.

The study specifically aims to extract relevant spatial and textural characteristics from digital images, develop a supervised classification model for estimating pixel-level embedding suitability, and design an adaptive embedding mechanism that adjusts the number of embedded bits according to classification results. Another objective is to examine the relationship between embedding capacity, perceptual distortion, computational complexity, and statistical detectability under different payload conditions [3]. The proposed contribution lies in integrating feature-based machine learning classification with dynamic spatial-domain embedding rather than applying identical LSB modifications across all available pixels. The framework is intended to provide a reproducible methodology for evaluating whether machine learning-guided pixel selection offers measurable advantages over conventional and manually thresholded adaptive LSB techniques.

## 3. Materials and Methods

### 3.1 Proposed System Architecture

The proposed system consists of five interconnected stages: cover-image acquisition, image-feature extraction, intelligent pixel classification, dynamic LSB embedding, and stego-image generation. During the first stage, a digital cover image is selected and converted into a suitable image representation for processing. The image is then divided into pixels or small local neighbourhoods from which spatial, statistical, and textural features are extracted. These features are supplied to a trained machine learning classifier, which estimates the embedding suitability of each candidate pixel or region. Based on the classification output, the dynamic embedding module determines the appropriate embedding decision and bit allocation. The secret message is subsequently converted into a binary sequence and embedded into the selected pixel locations through an LSB substitution mechanism. The resulting stego-image is evaluated against the original cover image to determine the degree of distortion and its suitability for secure information hiding. A corresponding extraction procedure uses the same classification and selection parameters, together with synchronized embedding information, to recover the embedded message.

*The general processing sequence is represented as follows:*

$$[\text{Cover Image} \rightarrow \text{Feature Extraction} \rightarrow \text{Machine Learning Classification}]$$
$$[\text{Classification Output} \rightarrow \text{Dynamic LSB Embedding} \rightarrow \text{Stego-Image}]$$
$$\text{Stego-Image} \rightarrow \text{Pixel Selection} \rightarrow \text{LSB Extraction} \rightarrow \text{Recovered Message}]$$

### 3.2 Cover-Image Acquisition and Preprocessing

The cover image represents the original digital image used to accommodate the confidential information. For experimental evaluation, a diverse collection of uncompressed or losslessly stored images should be selected to represent different visual characteristics, including smooth surfaces, textured regions, natural landscapes, buildings, objects, and complex structures. Images may be processed in grayscale or RGB format depending on the intended embedding strategy.

During preprocessing, the image dimensions, pixel intensity ranges, and colour-channel information are retained to ensure consistency between feature extraction, embedding, and recovery. Images should be stored in lossless formats such as PNG or BMP during the primary experiments because lossy compression may alter embedded LSB information and interfere with extraction accuracy. Where colour images are used, the embedding strategy should clearly specify whether classification is performed independently for each channel or through a combined representation of local colour and spatial features.

### 3.3 Image Feature Extraction

Feature extraction is an important stage of the proposed framework because the classification model depends on the quality and relevance of the information supplied to it. Instead of selecting pixels solely according to their intensity values, the proposed method considers multiple characteristics describing local image complexity and spatial relationships. Local variance is used to estimate the degree of intensity variation within a neighbourhood, while gradient magnitude represents the strength of local intensity transitions. Local entropy provides information about the diversity of pixel values, and neighbouring-pixel differences characterize spatial changes between adjacent pixels. Texture-related features can additionally be extracted using statistical descriptors or local texture operators. These characteristics collectively provide a feature representation that can help distinguish relatively smooth regions from complex or structurally varied areas.

*The local variance of a neighbourhood can be expressed as:*

$$[V(x,y) = \frac{1}{|\Omega|} \sum_{(i,j)\in\Omega} [I(i,j) - \mu_\Omega]^2]$$

where (I(i,j)) represents the pixel intensity, $(\mu_\Omega)$ denotes the local mean intensity, and (|\Omega|) represents the number of pixels within the neighbourhood.

Gradient magnitude can be estimated using horizontal and vertical image derivatives:

$$[G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2}]$$

where (G_x) and (G_y) represent the horizontal and vertical gradient components, respectively. These components may be calculated using Sobel operators or other established gradient-detection techniques.

Local entropy is calculated as:

$$[H = -\sum_{i=0}^{L-1} p_i \log_2 p_i]$$

where (p_i) denotes the probability of a particular intensity value and (L) represents the number of possible intensity levels. Neighbouring-pixel differences can be expressed as:

$$[D(x,y) = |I(x,y) - I(x+1,y)|]$$

The extracted features are organized into a feature vector:

$$[F(x,y) = [V, G, H, D, T]]$$

where (V), (G), (H), (D), and (T) represent local variance, gradient magnitude, entropy, neighbouring-pixel difference, and a selected texture descriptor, respectively. Additional features may be incorporated according to the image representation and computational requirements. Feature normalization should be performed where necessary to prevent variables with larger numerical ranges from disproportionately influencing the classification model.

### 3.4 Intelligent Pixel Classification Using Machine Learning

The proposed intelligent pixel-classification module uses supervised machine learning to estimate the suitability of individual pixels or image regions for information embedding. A Random Forest classifier is proposed as the primary classification model because it can process multiple numerical features, represent nonlinear relationships, and provide feature-importance estimates. The model receives the feature vectors extracted from the cover image and assigns each candidate pixel or region to a predefined embedding-suitability category. For the proposed framework, three categories are considered: low suitability, moderate suitability, and high suitability. Low-suitability regions generally correspond to relatively smooth or perceptually sensitive areas, whereas moderate- and high-suitability regions represent areas with increasing local complexity or variation.

The classifier can be represented as:

$$[C(x,y) = f_\theta[F(x,y)]]$$

where (C(x,y)) represents the predicted suitability class, (F(x,y)) denotes the extracted feature vector, (f_{\theta}) represents the trained classification model, and (\theta) denotes its learned parameters.
A probability-based classification mechanism can also be used:

$$[P_k(x,y) = Pr(C(x,y) = k \mid F(x,y))]$$

where (P_k(x,y)) represents the predicted probability that a pixel belongs to suitability class (k). The probability output can provide additional flexibility when selecting embedding locations because pixels with uncertain classification can be treated conservatively. For model development, training labels must be defined using a transparent and reproducible procedure. One possible strategy is to generate candidate embedding modifications on training images and assess the resulting local distortion and perceptual impact under a specified payload condition. Pixels or regions can then be assigned suitability labels according to predefined criteria. These labels should be generated independently of the final test images to avoid data leakage. The classification dataset should be separated into training, validation, and testing subsets at the image level rather than randomly distributing neighbouring pixels from the same image across all subsets. This separation is important because pixels from the same image may share highly correlated characteristics, potentially producing overoptimistic classification results if the dataset is divided only at the pixel level.

### 3.5 Dynamic LSB Data Embedding

Following pixel classification, the proposed dynamic embedding mechanism determines how secret information should be distributed across the selected image locations. Unlike conventional LSB substitution, which generally embeds one bit in every selected pixel or channel, the proposed method uses the predicted suitability category to control

embedding decisions. High-suitability regions may be assigned a greater embedding priority, moderate-suitability regions may receive limited embedding, and low-suitability regions may be excluded from embedding. For an initial implementation, a conservative dynamic allocation strategy can be adopted in which high-suitability locations accommodate one secret bit, moderate-suitability locations are used only when additional capacity is required, and low-suitability locations remain unchanged. A more extensive implementation may investigate controlled multi-bit embedding, provided that the additional distortion and detectability are evaluated separately. The dynamic embedding decision can be expressed as:

$$[b(x,y) = \{1, \quad C(x,y) = \text{High } 1, \quad C(x,y) = \text{Moderate and capacity is required } 0, \quad C(x,y) = \text{Low}]$$

where (b(x,y)) indicates whether a pixel is selected for embedding. The embedding operation for a selected 8-bit pixel can be represented as:

$$[P'(x,y) = (P(x,y) \quad 254) \mid m_j]$$

where (P(x,y)) represents the original pixel intensity, (P'(x,y)) denotes the modified intensity, and (m_j) represents the corresponding secret-message bit. The proposed strategy seeks to reduce unnecessary modifications by restricting embedding to locations considered suitable by the classifier. A pseudorandom permutation of the eligible locations, controlled by a shared secret seed or key, may additionally be used to avoid predictable sequential embedding. However, randomization should be treated as a complementary mechanism and not as proof of statistical security.

### 3.6 Secret-Message Preparation and Embedding Synchronization

Before embedding, the secret message is converted into a binary sequence using a defined character encoding, such as UTF-8. The binary payload should include sufficient information to identify its length and format during extraction. For practical secure communication, the message may also be encrypted before embedding, although the present study focuses on the intelligent pixel-selection and dynamic LSB mechanism. The selected pixel coordinates or their deterministic selection order must be reproducible at the receiver's end. This can be achieved by applying the same feature-extraction procedure and trained classifier to the received image, provided that the image has not undergone transformations that alter the relevant features. Alternatively, the embedding system may use a synchronized pseudorandom selection procedure with a shared secret seed and a predefined classification configuration. The payload length and required extraction parameters should be handled carefully because any mismatch between the sender and receiver may result in incomplete or incorrect message recovery.

### 3.7 Extraction Procedure

The extraction module retrieves the secret binary sequence from the received stego-image by reproducing the pixel-selection and embedding-order procedures used during transmission. Initially, the stego-image is loaded and subjected to the same preprocessing operations applied to the cover image. The feature-extraction and classification stages are then repeated, or the synchronized selection sequence is reconstructed using the agreed parameters. The least significant bits of the selected pixel values are subsequently extracted in the correct order and combined to reconstruct the embedded binary payload [4]. The recovered sequence is converted into its original message representation using the corresponding encoding procedure. If the payload was encrypted before embedding, an appropriate decryption operation must be performed after extraction. The accuracy of this stage depends on consistent image representation, synchronized classification parameters, correct payload-length information, and the absence of image transformations that disrupt the embedded bits.

## 4. Proposed Algorithm

The overall procedure of the proposed machine learning-assisted adaptive LSB method can be summarized in the following algorithm.

**Algorithm 1: Machine Learning-Assisted Dynamic LSB Embedding**

**Input:** $Coverimage(I), secretmessage(M), trainedclassifier(f_\theta).$

**Output:** $Stego - image(I_s).$

1. Load the cover image and convert it into the required image representation.
2. Convert the secret message into a binary sequence.
3. Extract local variance, gradient magnitude, entropy, neighbouring-pixel differences, and texture features.
4. Construct a feature vector for each candidate pixel or image region.
5. Apply the trained classifier to estimate pixel-embedding suitability.
6. Divide candidate locations into low-, moderate-, and high-suitability categories.
7. Select eligible embedding locations according to the dynamic allocation policy.
8. Apply a synchronized pseudorandom ordering to the selected locations when required.
9. Embed the secret bits using LSB substitution.
10. Preserve nonselected pixel values without modification.
11. Generate and save the stego-image.
12. Evaluate image quality, embedding capacity, extraction accuracy, and statistical detectability.

The extraction procedure reverses the embedding process by identifying the same eligible pixel locations, retrieving their least significant bits, and reconstructing the original secret message.

## 5. Experimental Design and Evaluation

The proposed method should be evaluated using a diverse collection of standard digital images representing different spatial and textural characteristics. The experimental dataset may include images from established image-processing collections, such as USC-SIPI, BOSSBase, or other appropriately licensed datasets.

The selected images should be divided into training, validation, and testing groups at the image level to prevent information leakage between the classifier-development and final evaluation stages [5]. The experimental study should compare the proposed machine learning-assisted method with conventional sequential LSB substitution and, where possible, a threshold-based adaptive LSB method. All methods should be tested using equivalent image dimensions and comparable payload sizes to ensure a fair performance comparison. The embedding process should be evaluated at multiple payload rates to determine how the proposed classification mechanism performs as the quantity of concealed information increases. Suitable payload levels may be selected according to the available embedding capacity of each image and reported consistently in bits per pixel. The primary evaluation metrics should include Mean Squared Error, Peak Signal-to-Noise Ratio, Structural Similarity Index Measure, embedding capacity, extraction accuracy, and computational execution time. Statistical steganalysis should also be incorporated to investigate whether the generated stego-images exhibit detectable characteristics.

### 5.1 Mean Squared Error

Mean Squared Error measures the average squared difference between the cover image and the stego-image and is calculated as:

$$[MSE = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}[I(x,y) - I_s(x,y)]^2]$$

where (I(x,y)) and (I_s(x,y)) represent the original and modified pixel values, respectively. Lower MSE values indicate smaller numerical distortion.

### 5.2 Peak Signal-to-Noise Ratio

Peak Signal-to-Noise Ratio evaluates the relationship between the maximum possible image intensity and the measured distortion:

$$[PSNR = 10\log_{10}\left(\frac{MAX_I^2}{MSE}\right)]$$

where (MAX_I) represents the maximum possible pixel intensity, typically 255 for an 8-bit image. Higher PSNR values generally indicate lower pixel-level distortion.

### 5.3 Structural Similarity Index Measure

SSIM assesses image similarity through luminance, contrast, and structural information. Its mathematical expression is:

$$[SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}]$$

where $(\mu_x) and (\mu_y)$ denote local mean intensities, $(\sigma_x^2)$ and $(\sigma_y^2)$ represent variances $(\sigma_{xy})$ denotes

covariance, and ($C_1$) and ($C_2$) are stabilizing constants. Values closer to 1 indicate greater structural similarity.

### 5.4 Embedding Capacity and Payload Rate

Embedding capacity represents the amount of information that can be concealed within the cover image under the selected embedding strategy. The payload rate can be expressed as:

$$[BPP = \frac{B_e}{WH}]$$

where ($B_e$) represents the total number of embedded bits and (WH) denotes the number of image pixels. For RGB images, the reporting convention should clearly distinguish bits per image pixel from bits per colour-channel component.

### 5.5 Extraction Accuracy

Extraction accuracy measures the proportion of embedded bits correctly recovered from the stego-image and can be expressed as:

$$[EA = \frac{N_c}{N_t} \times 100]$$

where ($N_c$) denotes the number of correctly recovered bits and ($N_t$) represents the total number of embedded bits. A value of 100% indicates complete recovery under the tested transmission conditions.

### 5.6 Computational Complexity

The computational efficiency of the proposed framework should be evaluated by recording feature-extraction time, classifier inference time, embedding time, extraction time, and total processing time. The additional computational requirements introduced by machine learning should be compared with the relatively low processing cost of conventional LSB substitution. This evaluation is important because improvements in pixel selection may not be practically beneficial if they introduce excessive processing overhead.

## 6. Comparative Evaluation Framework

The proposed approach should be compared with conventional and adaptive LSB methods to determine whether machine learning-assisted pixel classification provides measurable improvements. The comparison should consider not only image quality but also payload capacity, extraction reliability, computational complexity, and statistical detectability.

| Evaluation parameter | Conventional LSB | Threshold-based adaptive LSB | Proposed ML-assisted LSB |
|---|---|---|---|
| Pixel selection | Sequential or predefined | Feature-threshold based | Machine learning classification |
| Feature extraction | Generally unnecessary | Required | Required |
| Embedding allocation | Usually uniform | Rule-based | Classification-guided |
| Dynamic embedding | Limited | Possible | Central design feature |
| Computational complexity | Low | Moderate | Model-dependent |
| Adaptability to image content | Limited | Threshold-dependent | Learned feature relationships |
| Statistical security | Requires evaluation | Requires evaluation | Requires evaluation |
| Visual quality | Payload-dependent | Selection-dependent | To be experimentally established |

The proposed method should also undergo an ablation study to identify the contribution of individual components. For example, the performance of the classifier can be evaluated without entropy features, without gradient information, and without dynamic embedding allocation. Such comparisons can establish whether improvements arise from the machine learning model itself, the feature representation, or the embedding strategy.

## 7. Results and Discussion

The experimental results should be reported after implementation using the selected image dataset, trained classifier, and predefined payload conditions. Particular attention should be given to the relationship between pixel-classification performance and the final image-quality measurements. A high classification accuracy does not necessarily guarantee superior steganographic performance because the classification labels describe embedding suitability rather than directly establishing resistance to steganalysis [6]. Therefore, the model should be evaluated using both classification metrics and end-to-end steganographic measurements. Classification performance may be assessed through precision, recall, F1-score, and confusion matrices, whereas the resulting stego-images should be evaluated using PSNR, SSIM, MSE, and payload rate. The following table provides the recommended structure for presenting the experimental findings. Numerical entries should be populated using actual implementation results rather than estimated values [7-8]. The discussion should examine whether the proposed method reduces distortion at equivalent payload rates, whether the classifier consistently identifies suitable embedding locations across different image categories, and whether its additional computational cost is justified by the resulting performance. The effect of increasing payload rates should also be investigated because an embedding strategy that performs effectively at low payload levels may experience reduced imperceptibility when the available suitable locations become insufficient. Finally, statistical steganalysis should be used to examine whether the proposed embedding distribution remains distinguishable from original cover images. These findings will be essential for determining the practical value of machine learning-assisted pixel classification.

## 8. Security Considerations and Limitations

Although machine learning-assisted pixel classification can improve the flexibility of image embedding, it does not independently guarantee steganographic security. A classifier may identify perceptually complex regions, but these regions can still contain statistical characteristics that are sensitive to LSB modification. Consequently, high texture complexity or local variance should not be interpreted as evidence of resistance to statistical detection. The training dataset also plays an important role because a model trained on a limited range of image types may not generalize effectively to unseen images with different spatial or colour characteristics.

Furthermore, an embedding classifier may become biased if its training labels are generated using arbitrary thresholds or if training and testing samples contain highly correlated pixels from the same source images. Another limitation involves the computational overhead associated with feature extraction and model inference, particularly for high-resolution images and resource-constrained devices [9-11]. Dynamic allocation can also reduce usable embedding capacity when a large proportion of pixels are classified as unsuitable. These challenges indicate that the proposed framework requires careful training-data construction, image-level validation, payload control, and independent steganalysis experiments before its security advantages can be established.

## 9. Future Research Directions

Future development of machine learning-assisted image steganography may investigate more advanced classification architectures, including gradient-boosting models, support vector machines, convolutional neural networks, and lightweight vision-based models. These approaches may improve the ability to identify complex relationships between spatial features and embedding suitability. Another important direction involves developing cost-aware classification models that jointly consider visual distortion, payload requirements, and statistical detectability rather than relying exclusively on texture or edge characteristics. Researchers may also investigate reinforcement-learning-based embedding policies in which the selection strategy is optimized according to predefined image-quality and capacity objectives. Furthermore, incorporating cryptographic encryption before embedding can provide confidentiality protection if an unauthorized party extracts the hidden payload. Robust extraction under image transformations, including resizing, filtering, and compression, represents another important research challenge. However, any improvement in robustness must be assessed against its potential effects on embedding capacity and imperceptibility. Finally, reproducible datasets, image-level train-test separation, open implementations, and standardized steganalysis protocols will be necessary to establish meaningful comparisons between machine learning-assisted and conventional steganographic techniques.

## 10. Conclusion

This study proposes a machine learning-assisted image steganography framework that integrates intelligent pixel classification with dynamic LSB data embedding to address limitations associated with uniform pixel modification. The proposed architecture extracts spatial and textural characteristics, including local variance, gradient magnitude, entropy, and neighbouring-pixel differences, and uses a supervised classification model to estimate the embedding suitability of individual pixels or image regions. The classification output guides the dynamic allocation of secret information, allowing selected regions to accommodate the payload while reducing unnecessary modifications in less suitable areas.

The framework establishes a systematic methodology for comparing machine learning-assisted embedding with conventional and threshold-based LSB techniques using image-quality, capacity, extraction, computational, and statistical detection measurements. However, the effectiveness of the proposed method must be established through implementation and controlled experimentation, as intelligent pixel classification alone does not guarantee statistical security or superior image quality. Future work should therefore focus on validating the classifier across diverse image datasets, optimizing the dynamic embedding strategy, and evaluating the resulting stego-images against advanced steganalysis techniques. The proposed framework provides a foundation for further investigation into intelligent, content-adaptive, and computationally efficient image steganography.

## References


1. Anderson, R. J., & Petitcolas, F. A. P. (1998). On the limits of steganography. *IEEE Journal on Selected Areas in Communications, 16*(4), 474–481. https://doi.org/10.1109/49.668971
2. Chan, C. K., & Cheng, L. M. (2004). Hiding data in images by simple LSB substitution. *Pattern Recognition, 37*(3), 469–474. https://doi.org/10.1016/j.patcog.2003.08.007
3. Fridrich, J., Goljan, M., & Du, R. (2001). Detecting LSB steganography in color and gray-scale images. *IEEE Multimedia, 8*(4), 22–28. https://doi.org/10.1109/93.959097
4. Hussain, M., Wahab, A. W. A., Idris, Y. I. B., Ho, A. T. S., & Jung, K. H. (2018). Image steganography in spatial domain: A survey. *Signal Processing: Image Communication, 65*, 46–66. https://doi.org/10.1016/j.image.2018.03.012
5. Johnson, N. F., & Jajodia, S. (1998). Exploring steganography: Seeing the unseen. *Computer, 31*(2), 26–34. https://doi.org/10.1109/2.657957
6. Luo, W., Huang, F., & Huang, J. (2010). Edge adaptive image steganography based on LSB matching revisited. *IEEE Transactions on Information Forensics and Security, 5*(2), 201–214. https://doi.org/10.1109/TIFS.2010.2041812
7. Mielikainen, J. (2006). LSB matching revisited. *IEEE Signal Processing Letters, 13*(5), 285–287. https://doi.org/10.1109/LSP.2006.870357
8. Pevný, T., Bas, P., & Fridrich, J. (2010). Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security, 5*(2), 215–224. https://doi.org/10.1109/TIFS.2010.2045842
9. Qian, Y., Dong, J., Wang, W., & Tan, T. (2015). Deep learning for steganalysis via convolutional neural networks. In *Proceedings of SPIE 9409, Media Watermarking, Security, and Forensics 2015*. https://doi.org/10.1117/12.2083479
10. Westfeld, A., & Pfitzmann, A. (2000). Attacks on steganographic systems. In A. Pfitzmann (Ed.), *Information hiding* (Lecture Notes in Computer Science, Vol. 1768, pp. 61–75). Springer. https://doi.org/10.1007/10719724_5
11. Yang, C. H., Weng, C. Y., Wang, S. J., & Sun, H. M. (2008). Adaptive data hiding in edge areas of images with spatial LSB domain systems. *IEEE Transactions on Information Forensics and Security, 3*(3), 488–497. https://doi.org/10.1109/TIFS.2008.926097