

Secure Data Exchange: An Integrated Approach Using Convolutional Neural Networks and Steganography on MNIST Digital Number Images

Chrakhan H. Karim

Department of Statistics and Informatics, College of Administration and Economics, University of Sulaimani, Sulaymaniyah, Kurdistan Region, Iraq

Email: chrakhan.hassan@univsul.edu.iq

Miran H. Mohammed

Department of Basic Science, College of Dentistry, University of Sulaimani, Sulaymaniyah, Kurdistan Region, Iraq

Email: miran.mohammed@univsul.edu.iq

ARTICLE INFO

Article History:

Received: 13/2/2025

Accepted: 6/4/2025

Published: Spring 2026

Keywords:

Steganography, Data security, MNIST, CNN, Cryptography

Doi:

10.25212/lfu.qzj.11.2.42

ABSTRACT

With the widespread use of digital communication, secure data transfer has become a top priority nowadays, which is important to prevent data leaking and hacking. This paper introduces a new concept that combines steganography and deep learning to protect confidential data during transfers. It uses the Modified National Institute of Standards and Technology (MNIST) dataset of handwritten digits to create a steganographic system that hides the required text within the images; before the steganography process, the text is encrypted with a cryptography technique, and for better security, a new approach of securing text is used, which make the encrypted text undetectable, and this technique adds additional level security, especially when it is merged with steganography. Also, this steganographic technique is integrated with a Convolutional Neural Network (CNN), resulting in a dual-purpose model that both disguises and predicts the presence of hidden messages with an impressive 99% accuracy. This method ensures that only individuals with the correct decryption key can access the confidential information, thereby ensuring the security of the transferred data. The fusion of these two methods sets a new standard for secure data exchange in modern

networks, offering a robust framework that mitigates the vulnerabilities of current communication systems. Consequently, the detection of image-based steganography is highly accurate and significantly contributes to the development of data protection in the face of emerging artificial threats.

1. Introduction

With the widespread use of cloud computing and web technology, data security has emerged as a critical concern across multiple industries. The escalating number of data breaches and vulnerabilities in cloud storage has underscored the pressing need for robust data protection and communication protocols. In this era of rampant cyber-attacks, the importance of data protection cannot be overstated, making it a top priority for individuals, organizations, and governmental institutions. (Alanzzy et al., 2023).

The interconnected world has made digital interactions between individuals, non-governmental organizations, and governmental institutions the norm. This heavy reliance on technology has led to a global surge in cyber-attacks, underscoring the urgency and significance of the issue. These attacks, ranging from viruses to data breaches, pose a significant threat to our digital lives. However, the solution for cyber-attacks is not a task that a single entity can undertake. Mitigating the impact of network attacks necessitates a collective effort, and numerous solutions are being devised worldwide. It is imperative that each individual and organization recognize their role in finding solutions for security attacks. This study presents a repertoire of security techniques to combat any digital threats, underscoring the power and responsibility of every stakeholder in this battle. (Li & Liu, 2021).

In the digital world, the first priority, physical or electronic, is to ensure a secure environment for hiding records of sensitive information. Thus, in this context, providing secure storage and communication of confidential and personal information has become essential to our professional and personal lives. Some techniques, such as cryptography, steganography, and other security techniques, have proven to be more effective in securing personal and organizational data from cyber-attacks (Majeed et al., 2021).

The primary objective of this paper is to provide a secure and efficient method for data exchange over a network by using different techniques, such as cryptography for data encryption, steganography for embedding encrypted data, as well as collaborating with the use of the Convolutional Neural Network (CNN) model for predicting the image with embedded encrypted text. This study also uses the Modified National Institute of Standards and Technology (MNIST) handwritten digit dataset, which consists of nearly 60,000 images, and these images are used to embed the text that needs to be hidden.

This paper highlights several significant points. Firstly, it points out the critical need for advanced data security solutions in an era where cyber-attacks are becoming increasingly sophisticated. The second point is that it combines two techniques, the CNN model with steganography, which not only enhances security but also ensures that the data remains undetectable by unauthorized parties. Lastly, the findings of this paper may have implications for providing a secure communication system, protecting personal data, and safeguarding critical infrastructure. As a result, this study proposes a new approach to data exchange security by integrating different methods using cryptography and steganography MNIST images for convolutional neural networks. Thus, it could open the way to new security applications that combine the strengths of cryptography and steganography and potentially reduce the risks associated with data breaches and cyber-attacks.

The paper is arranged as follows. It begins with a general introduction, addresses the main problem that this paper tries to address, and presents the proposed solution. Then, it gives a general overview of the material and methods that are used in this paper to give general information about the technique that this paper is trying to follow. After that, the previous work done in the same area will be demonstrated, and the following methodologies and limitations will be addressed. After that, the methodologies of the paper will be discussed, including descriptions of the models and security techniques in the encryption and decryption parts followed by the CNN model. We then present a comprehensive analysis of results obtained during the encryption and decryption process and evaluate the prediction accuracy by the CNN model. Finally, we conclude the paper and discuss future work.

2. Literature Review

2.1 Cryptography Techniques

Cryptography has played an important role in war and peace, dating back to ancient civilizations such as Egypt and Greece, which used primitive substitution ciphers, to innovative developments during World War Two (WWII), particularly the Enigma machine and the work at Bletchley Park. Techniques progressed from simple substitutions to complicated algorithms such as RSA and AES (Advanced Encryption Standard), ushering in the digital era of secure communication and data security. However, as cryptography advanced, so did the art of cryptanalysis, or the science of breaking codes and ciphers, resulting in a never-ending cat-and-mouse game between code makers and codebreakers that continues to shape the discipline today (Dooley, 2018).

Cryptography applies mathematical techniques and algorithms to encrypt and decrypt data. Encryption involves converting plaintext data into unreadable ciphertext using an algorithm and key, while decryption reverses this process to retrieve the original plaintext. The primary purpose of data encryption is to ensure the confidentiality, integrity, and protection of sensitive information against unauthorized access during transmission over insecure communication channels. The sender encrypts the data and transmits it to the receiver, who then decrypts the ciphertext to obtain the plaintext (Mandal et al., 2014).

Cryptography can be divided into two categories based on the number of keys involved: symmetric and asymmetric. Symmetric key encryption is called single-key, private key, or conventional encryption. The sender and receiver use the same key for encryption and decryption in this system. However, the key must be securely exchanged between the two parties. Symmetric key algorithms guarantee the confidentiality and integrity of data. Examples of symmetric key algorithms are Data Encryption Standard (DES), 3DES, Blow-fish, and AES (Chandra et al., 2014)

Asymmetric key encryption, also known as public key encryption, entails using two separate keys for encryption and decryption purposes. These keys are commonly referred to as the public key and the private key. Public key encryption guarantees the secure transmission of data. Prominent examples of public key algorithms include

RSA, Diffie-Hellman, Elliptic Curve Cryptography (ECC), and Digital Signature (Chandra et al., 2014).

2.2 Image Steganography

“A Comprehensive Review on Advancements and Applications of Steganography” presents a historical overview of steganography, tracing its origins to ancient Greece, where messages were hidden on wax tablets or tattooed on messengers' bodies. It emphasizes the evolution over the ages, such as the employment of invisible ink in the Renaissance and microdots during WWII. The study also discusses the emergence of modern digital approaches, in which data is hidden within multimedia files such as photos, audio, and video, emphasizing how steganography has constantly adapted to technological improvements to ensure safe communication (Divyasree et al., 2024). Steganography conceals secret multimedia data, such as images, text, files, or movies, within a larger multimedia file. This concealed data is undetectable to the naked eye; thus, it does not raise suspicion. Unlike encrypted data, which seems chaotic and can draw attention, steganographic data blends in with the host file, remaining unremarkable (Subramanian et al., 2021).

2.3 Convolutional Neural Networks CNN-based Techniques

One important network in machine learning is the CNN. CNNs have been crucial in achieving breakthroughs in image and text analysis. They have gained attention in industry and academia. However, there is a lack of understanding regarding their application in data hiding, even though neural network-based data hiding is a significant area of research. With advancements in machine learning, CNNs are becoming more complex and diverse (Li et al., 2022) (Dzhanashia & Evsutin, 2024). A CNN model consists of three layers: input, hidden, and output. The hidden layers typically include convolutional and pooling layers, which perform convolutions. The combination of these layers forms the fundamental architecture of a CNN, which has proven effective in image classification, object detection, and image segmentation (Sony et al., 2021).

2.4 MNIST Datasets

The MNIST, is a standard dataset in machine learning and computer vision commonly used for image classification tasks. Researchers extensively utilize the MNIST dataset in academic research to develop and test new machine-learning algorithms. Its primary purpose is to train models to recognize the numbers represented in each image accurately. MNIST comprises ten classifications (0 through 9), with each handwritten digit being 28x28 pixels. The size of the image does not impact the variations of the numbers within the dataset. Therefore, even a small image can have the same number of variations as a large image (Rana et al., 2023)

2.5 Grayscale Images

The use of grayscale images in CNN research offers several advantages. Grayscale images contain intensity information instead of color, which reduces the model's input dimension and computational complexity. This is particularly useful when computing resources are limited or when the goal is faster training and inference. In addition, using grayscale images allows researchers to focus on the images' internal texture and structural properties, simplifying the model learning process. Moreover, in certain fields, such as medical imaging or satellite imaging, color may not always convey important information, and significant details can often be adequately represented in grayscale. Thus, selecting grayscale images in CNN research facilitates more efficient and specialized model training, which in turn improves interpretability and potentially model performance for certain tasks (Tomasi & Manduchi, 1998).

2.6 ASCII code

ASCII code stands for American Standard Code for Information Interchange, a character encoding standard that assigns numerical values to a set of 128 characters, including letters, numerals, punctuation marks, and control characters. Each character is represented by a unique 7-bit binary code, allowing computers to exchange and interpret textual information. ASCII is a fundamental encoding system for text-based communication widely used in computing and telecommunication applications (Karaçam et al., 2021).

2.7 Related Work

(Alanzay et al., 2023) discuss the idea of a multi-level steganography (MLS) algorithm that uses AES and Blow-fish to protect the cover image and its encryption key as the key images of the stego image. The study aims to use the algorithm named MLS, which uses the Least Significant Bit (LSB) method that functions by randomizing pixels to enhance security, the MLS method also contains a secret key derived from an image and divides the image into smaller blocks.

This paper (Li et al., 2020) presents a grayscale steganographic approach to hide an encrypted secret image as a cover image of the same size. Before hiding, the image is encrypted using chaotic encryption technology. Next, the encrypted and cover images are converted into brace images using a CNN model. In addition, a generative adversarial network (GAN) is used. The steganography system consists of three CNNs: hidden, discriminative, and discriminative networks. A weight distribution mechanism is also needed to ensure balanced training of the hidden networks.

(Kich and Bachir, 2020) proposes an image steganography system that uses auto-encoding networks based on end-to-end trained deep convolutional neural networks for image hiding and extraction. The sender uses the network to encode the color image into another photo of the same size; the recipient uses another subnet to obtain the secret image of the received brace.

(Priya et al., 2024) develop deep neural networks to combine two-dimensional color images of the exact resolution into a single cover image. The embedded cover image is encrypted by combining a 3D chaotic map for diffusion and ECC for confusion.

This research (Yu et al., 2023) Introduces a coverless image steganography framework called Controllable, Robust, and Secure Image Steganography (CRoSS). This framework is the first attempt to introduce diffusion models in the field of image steganography.

(Li et al., 2024) develop an image-style-transformation neural network with integrated steganography functionality to embed hidden information in generating art-style images produced by artificial intelligence-generated content (AIGC). The scheme takes images as inputs (Content image, art-style image) alongside a message to be processed using an encoder-decoder model. The final generated image harbors the secret message.

(Long et al., 2024) propose an encryption system based on a convolutional neural network which is utilized to create an end-to-end medical image encryption system which uses the encoding and decoding abilities of deep learning to both encrypt and decrypt images. Feature keys are generated by chaotic systems fused with encoded plaintext image features to enhance security. Furthermore, reversible neural networks are used to encrypt dynamic fused parameters, which are then transmitted as encrypted parameter to improve reconstructive effectiveness.

This paper (Sanjalawe et al., 2025) proposes a multi-layered information hiding framework. It combines Huffman coding to compress data and mask statistical patterns. LSB embedding for efficient data hiding, and a deep learning-based encoder-decoder to provide an additional layer of protection by hiding one image within another. Extensive tests on benchmark datasets (Tiny ImageNet, COCO, and CelebA) demonstrate that this process achieves high visual fidelity (SSIM above 99%), robust data recovery, and improved resistance to attacks such as obfuscation and compression.

This study (Hossain et al., 2024) presents a convolutional neural network CNN-based steganographic model designed to hide multiple secret images within a single cover image. The model utilizes a multi-scale encoder and key-based decryption for improved safety. It successfully encodes and decodes two and three secret images with great precision. For two-image steganography, the encoded cover image reached 95.16% accuracy, while decoding accuracies reaching 97.16% and 97.28% for secret images 1 and 2, respectively. In three-image steganography, the cover image had 93.84% accuracy, with deciphered hidden images achieving 96.39%, 94.82%, and 94.76%. Security was additionally strengthened using AES and ChaCha20 encryption.

3. Materials and Methods

3.1 Python Programming Language

The programming language used in this research is Python. It is simple to use and program. the Google Collab iCloud service is used to execute the Python script. It is a platform in the cloud that enables you to write, execute, and share codes. It focuses on Python and operates within a Jupyter Notebook environment. Google hosts this platform, which offers a user-friendly setup for tasks such as data analysis, machine

learning, deep learning, and other programming activities (Bisong, 2019). Since the primary focus of this research paper involves working with images, particularly within CNN model architecture, high-quality memory and GPU are essential. Google Colab provides sufficient memory and GPU resources, and as such facilitating efficient image classification and prediction processes.

In addition, most of the required Python libraries are already installed in Colab, and it is not necessary to take another step to install them. The main libraries used in Python in this study are OpenCV, TensorFlow, Sklearn, Seaborn, and Matplotlib. OpenCV is used for image reading and processing. In contrast, TensorFlow is used to create CNN models for image prediction using Sklearn for dataset splitting and Seaborn with Matplotlib to visualize the performance of model predictions.

Additionally, Google Drive is used in conjunction with Google Colab for handling images and image datasets. The image dataset is stored in a folder on Google Drive, and these images are then loaded from that folder into the proposed CNN model for predictions.

3.2 CNN Model Design

The first step in developing the secure steganography approach is to create the architecture for the CNN used for image classification.

The images used in this study are a known dataset called MNIST, which consists of digital numbers from 0 to 9 written in different shapes and directions. This research utilizes the handwritten numbering Digit MNIST because it is simple and free from online datasets, and the CNN model uses them in conjunction with steganography techniques. These two processes are crucial for the security process—steganography embeds the message into the handwritten digit number image. At the same time, CNN predicts the number on the image that contains the hidden text among its pixels. Before the process of CNN model creation, it is required to start with reading images from the dataset folder in Google Drive, and for this, the OpenCV libraries are used. The images are read by using the `cv2.imread` function. The images are fetched in Red, Green, and Blue (RGB) color format, and they need to be converted to grayscale using the color function named `cvtColor`, which is a built-in function in Python, as the source of the images is required to be in this format. These images walk through the process

of downscaling, which is called the normalization technique. This technique divides each pixel from the image by 255, which is applied to all the images. Therefore, the size of images will decrease, and the training process of the CNN model will not suffer from prolonged time consumption, making the process faster.

The dataset used in this study is divided into training and validation sets, with 70% of the data allocated for training and 30% for validation. The training set is used to fit the model's parameters, while the validation set evaluates the model's performance during training, allowing adjustments to reduce overfitting and improve generalization. This splitting strategy helps assess the model's performance on unseen data, providing a more accurate measure of its effectiveness.

The process of splitting data into training and validation sets is considered one of the important steps in the CNN, which increases the performance evaluation of the model and generalizes it. The training part is used to optimize the model and evaluate how the model performs with accuracy, while the validation set is used to see how the model performs on unseen data. The main importance of this process is to reduce the rate of overfitting. This usually occurs when the dataset performs well on the training set and worse on the testing set. In addition, the efficiency of the validation set is to generalize the model to perform well on real-world data and avoid overfitting during the training process (Lecun et al., 1998).

Moreover, the augmentation technique is used because images might come in different sizes and shapes. Thus, this technique makes the model learn and extract more features and patterns from images and increases the number of image datasets, which is important to prevent overfitting from occurring. The main parameters in the augmentation process are zooming, shifting, and rotation within a range of values.

After all the images are ready, the next step starts with creating the architecture model, which consists of four convolve layers with 64 filters in size of 3 by 3. These filters extract important features from the MNIST image, such as edges, lines, and curves. Then, the batch normalization layer is used to normalize the input image, and it is useful for normalizing the image size and number of images as input to the model. Within each convolve layer, max-pooling is also used to reduce the size of the produced extracted feature, and in each layer, the RELU activation is used to learn

complex models. After max-pooling, the dropout layer drops inactivated neurons, reducing the overfitting effect.

Dropout layers are employed within the convolutional and dense layers to lessen overfitting and improve generalization. among the convolutional layers, a dropout rate of 0.25 is applied, demonstrating that neurons are randomly deactivated during training with an amount of 25%. This technique helps prevent the co-adaptation of neurons, contributing to a more robust model.

Another factor to reduce the risk of overfitting is using ReLU activation and dropout in the CNN model design. Moreover, the combination of ReLU activation and dropout provides effective feature extraction in the convolutional layers and enhances the reliability of the dense layers. The ReLU function is defined as the Equation 1.

$$f(x) = \max(0, x) \quad (1)$$

where **f(x)**: indicates the output function, which depends on input x values.

x: is the input to the function, where x is the output from the previous CNN layer and is used with ReLU.

max (0, x): this operation returns the maximum output of two values 0, x; if x is positive or zero, the function returns x; if x is negative, it returns 0.

After the CNN model is created, the upcoming layer is called the flattened layer; in this layer, all the extracted features from the previous layer are flattened into a 1-D array and to Artificial Neural Network (ANN) architecture, which works to predict the number of digits. There are two dense layers of 128 filters in the first hidden layer and 64 in the second hidden layer.

The last layer, called the output layer, predicts the final output of the digit number that is shown on the image, producing ten outputs. The SoftMax activation works by choosing the highest probability rate among the ten produced outputs.

However, the most important activation function is used as a neural network output layer. It is named SoftMax and is used for multi-class classification applications. This activation layer's main functionality is converting raw scores, or logits, into a probability distribution score over several classes. Each class receives a probability rate, which signifies the chance that the input belongs to that class. The SoftMax activation has also ability to translate the model outcome as a prediction value into interpretable probabilities, allowing for easier decision-making, and it uses cross-

entropy during the training evaluation process (Bishop, 2006). SoftMax equation It is defined as the Equation 2.

$$\text{SoftMax}(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2)$$

Where e^{z_i} : standard exponential function for the input vector.

e^{z_j} : standard exponential function for output vector.

K : number of classes in the multi-class classifier.

The requirement for training the CNN model on images is to optimize the balance of model speed and generalization of the model. This research uses the Adam optimizer, which is known for its ability to adapt the learning rate and handle complex optimization during the training process. The Adam optimizer also provides a faster convergence over the momentum and decreases the overfitting. Furthermore, the combined effect between Adam and the dropout layers contributes to a robust training process (Kingma & Ba, 2014).

Another important strategy is used along with the Adam optimizer, which is called the dropout technique; it serves as a regularization strategy by randomly deactivating a subset of neurons during training, and it avoids neuron co-adaptation, which reduces overfitting (Kingma & Ba, 2014). The Equation 3 shows Adam's adaptive learning rates:

$$\begin{aligned} m_t &= \beta_1 \times m_{t-1} + (1 - \beta_1) \times g_t \\ v_t &= \beta_2 \times v_{t-1} + (1 - \beta_2) \times g_t^2 \\ m^t &= \frac{m_t}{1 - \beta_1^t} \\ v^t &= \frac{v_t}{1 - \beta_2^t} \\ \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{v^t + \epsilon}} \times m^t \end{aligned} \quad (3)$$

where m_t : this is the “first-moment estimate” or the exponentially decaying average of past gradients.

β_1 : A hyperparameter that controls the decay rate for the first moment.

m_{t-1} : The first-moment estimate from the previous time step.

g_t : The gradient of the loss function with respect to the parameters at time step t .

v_t : This is the “Second-moment estimate” or the exponentially decaying average of past squared gradients.

β_2 : A hyperparameter that controls the decay rate for the second moment.

v_{t-1} : The second-moment estimate from the previous time step.

gt^2 : The square of the gradient at time step t .

m^t : This is the bias-corrected first-moment estimate in the early stage of training.

$1 - \beta_1^t$: Factor used to correct the bias in m_t

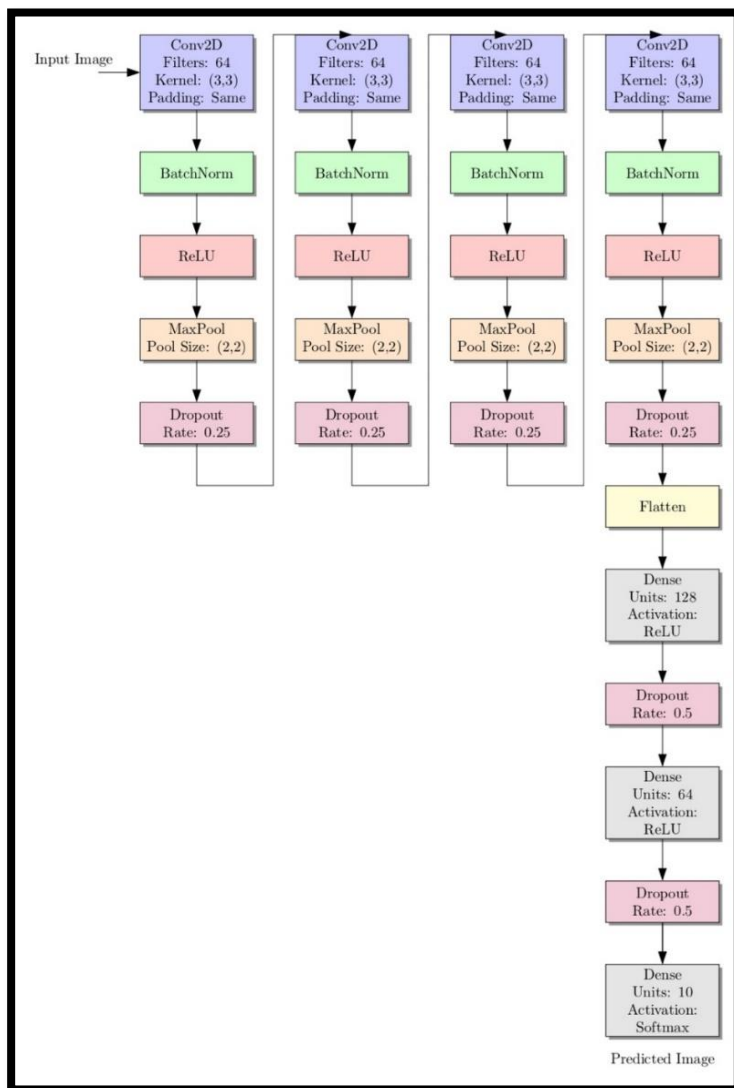
$1 - \beta_2^t$: Factor used to correct the bias in v_t

θ_t : The weight of the model at time step t .

η : Learning rate control the step size during learning and optimization.

ϵ : A constant value added to the equation to prevent division by zero.

Epochs represent a single complete process across the training dataset, which is important for model convergence and allows incremental adjustment of weights to capture complex patterns in complex data. Correctly setting the number of epochs is crucial to avoid under- or over-fitting, thus ensuring optimal model performance. Batch normalization is a technique that normalizes the layer inputs during training, stabilizes and speeds up the training process, reduces internal covariate biases, and enables higher learning rates. It also acts as a kind of regularization that improves generalization and facilitates the training of deeper networks. The synergy between epochs and group normalization promotes more stable, faster, and better generalized neural models (Ioffe & Szegedy, 2015). The designed model is trained on 100 epochs, with each completed epoch representing one complete iteration through the training data. On the other hand, the batch size, set at 64, determines the number of training image samples used in one iteration before the model's parameters are updated for the next epoch. This approach is important because it balances training speed and computational efficiency, providing a robust framework for the training process. The reason for developing the CNN model at the first stage is to prepare it for the next stage, which is used with steganography images. Thus, the CNN model must be used first. The reason is that the model waits for the input image as a handwritten digit and also contains the embedding of encrypted text messages using the steganography technique. Figure 1 shows the architecture model of the proposed CNN model.



3.3 Steganography and Cryptography encryption technique

The following process is the technique used to hide a text message in the handwritten digit image; in this process, cryptography and steganography strategies are used. The first step is to prepare the image environment and the nature of the inserted image. This step is a user manual image, writing a single digit number on a piece of paper by hand from 0 to 9, and it may take different shapes based on the handwritten quality

The paper is then scanned into a computer as a jpg or png file, converting the physical document into an electronic image format.

After the image is scanned, the OpenCV library is used to read it. Since the image originates from a physical document, it will likely contain noise artifacts. Thus, removing the noise and converting the image into a grayscale channel is essential. Therefore, the bilateral noise reduction technique is used in Python to remove the noise from the image and the OpenCV library function. It is used to convert color from RGB, which consists of three channels (red, green, blue), to grayscale as one channel. Bilateral noise reduction is a technique employed in image processing to reduce noise while preserving edges and details. It applies a weighted average to each pixel, considering spatial proximity and intensity similarity. This is achieved by combining spatial and intensity-dependent Gaussian filters, ensuring that neighboring pixels contribute more to the average while giving less weight to those with significantly different intensities (Tomasi & Manduchi, 1998). The next stage involves inserting the text message that needs to be sent over the network. Before embedding it in the image, the message should be reversed and encrypted. To achieve this, a proposed technique is used to convert each character of the reversed text to its ASCII code. Every character, whether uppercase or lowercase, has a unique ASCII number. The conversion from a character to its ASCII code is done using function `ord()`. For additional security, each ASCII code number is added by 1.

After that, the code number of each character is summed with that digit written on the inserted image. In this step, the result will be a list of integer values, and the length is equal to the number of characters of the inserted text.

The next step is inserting a digit number written on the inserted image as text. Each character is converted to an ASCII code number, and the result is saved in the new list. This process is done to combine the ASCII numbers of the digit text with the code numbers of the previously inserted text, resulting in a new text.

To enhance the security of the text, the resulting list from the previous step is converted to binary using the `format(ord(i), '08b')` function in Python programming. This produces a new list of binary numbers. Similarly, each pixel of the image must also be converted into binary format. Without this conversion, the text cannot be

hidden within the image. Therefore, at this stage, both the text and the image pixels should be in binary form.

In the final stage, the final form of calculated binary numbers in the list should be paired and converted to decimal numbers in pairs of tuples, where each tuple represents the greyscale of the pixel color of the image. Thus, the final output will be the greyscale image containing the number and a hidden text message inside, especially among the pixels.

This research uses the Least Significant Bits (LSB) technique for embedding. It refers to the rightmost bit in a binary representation of a number, representing the most minor or least significant value in that binary digit. In steganography, LSB manipulation involves replacing the least significant bit of pixel values in an image with bits of hidden information. This method exploits the fact that small changes in the LSB are less likely to be noticeable to the human eye, allowing for the concealment of additional data within the image. In digital systems, the significance of a bit decreases from left to right, with the LSB holding the most negligible weight (Begum & Uddin, 2020). Equation 4 is used for embedding using the LSB technique.

$$I'(x, y) = \left(\frac{I(x, y)}{2} \right) \times 2 + B(i) \quad (4)$$

Where $I(x, y)$: The value of a pixel at coordinate (x, y) in the original image I . For a greyscale image, the value ranges from 0 (black) to 255 (white).

B(i): This value represents a bias added to the CNN model to shift the result by a certain amount, and I is a particular feature.

Until this step, the prediction model, such as SNN and encryption techniques, like hiding text in images using steganography, have been established. Thus, the process of hiding text in handwritten images is completed. The next step is to create a decryption process on the receiver side. Figure (2) shows the flowchart of the encryption process.

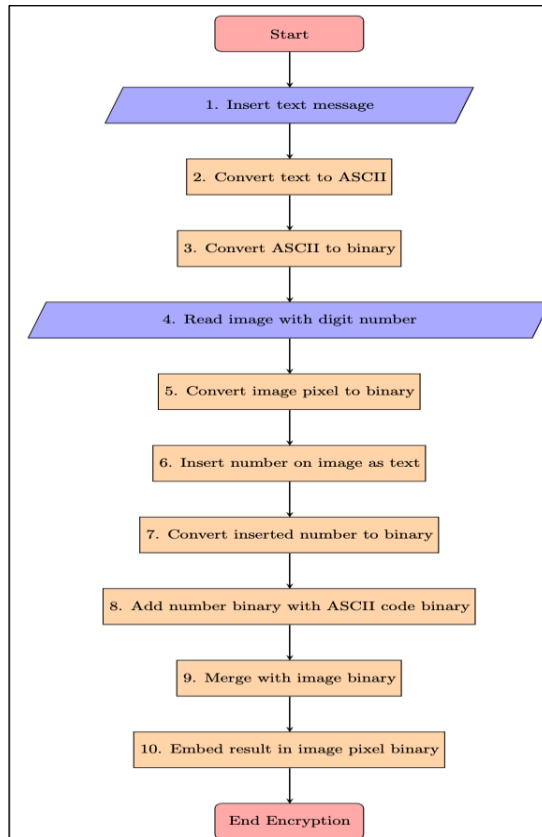


Figure (2): Encryption flowchart

3.4 Prediction Process Using Proposed CNN Model

The first decryption step starts with inserting the received image from the sender containing hidden text into the established CNN model to predict the number depicted in the image. This step goes through a set of epochs, typically 100 epochs. Furthermore, the model is tested with unseen data by using different shapes and orientations of handwritten digits on paper and applying them to the model to ensure accurate predictions.

3.5 Steganography and Cryptography decryption technique

After predicting the written number on the image, the reverse operation of LSB begins. In this operation, the main part extracts values from pixels of the image, especially those that likely hide the text message. However, the actual message will not be immediately visible after extraction, as it has undergone further operations during encryption. The Equation 5 is used to extract bits from the image.

$$B(i) = I(x, y) \bmod 2 \quad (5)$$

Where **I(x,y)**: is a particular feature of the image *I* at a particular coordinate (x,y) modular by 2.

After extracting information from the image, it is important to mention that in the encryption step, the text message is merged with the number text on the image; thus, it is necessary to isolate them from each other. This is achieved using the XOR equation, which extracts the complete text from the image containing both texts. The key idea is that, with proper authorization, the decryption key is known on the decryption side. Thus, the XOR operation is used to extract each text individually. The Equation 6 illustrates the XOR technique used to separate the two texts (Text message and Text Number).

$$N(i) = C_{\text{extracted}}(i) \text{ XOR } M(i) \quad (6)$$

$$M(i) = C_{\text{extracted}}(i) \text{ XOR } N(i)$$

Where **N(i)**: Its value of pixel of the input image layer.

C_{extracted}(i): This represents the extracted feature from the input image, such as edge, or texture in the the image.

M(i): This value used the encryption technique or data hiding over pixel value for the extracted features.

The outcome of the previous step produces two lists in ASCII code decimal number format: one for text messages and the other for numerical text. The values in the text message list are subtracted from those in the numerical text list, creating a new list of integer values. These new values are then subtracted from the predicted number,

and the results are compiled into another list. Each value in this list is subsequently decreased by 1, resulting in a list of numbers representing ASCII codes. The final step is to convert each of these numbers back into their corresponding characters. In the final step of encryption, the inserted text is reversed to disrupt the sequence of characters. Consequently, it needs to be reversed again to retrieve the original text inserted by the sender. Figure 3 shows the flowchart of the decryption process.

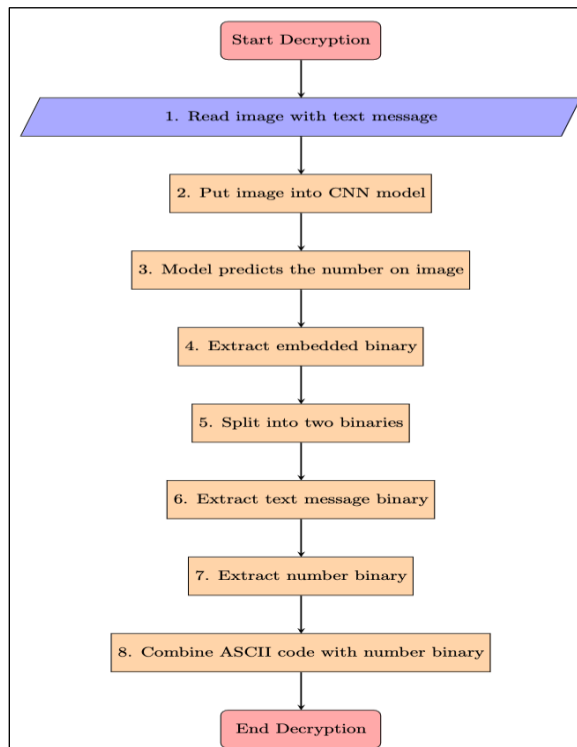


Figure (3): Decryption flowchart

4. Results and Discussion

4.1 Results

The proposed method for enhancing security using cryptography, steganography, and CNNs has been successfully implemented and rigorously tested. Using CNNs for image classification is key to this method, with the MNIST dataset proving particularly effective for recognizing handwritten digits. Furthermore, steganography techniques

have been used to disguise textual information within the images. This comprehensive methodology attempts to improve data security in digital communication channels by integrating advanced image analysis with covert data embedding techniques.

The research outcome shows significant results and accuracy performance. Consequently, these findings ensure a secure environment for transmitting messages over a network without the risk of data leakage by intruders.

4.2 CNN Model Architecture

In the CNN model, the architecture and hyperparameters used show high accuracy as predictions with an accuracy of 0.99%. This result indicates that the model is highly effective at predicting any image containing a handwritten number before it is scanned into the model. Additionally, while accuracy is often presented as the final result, it alone is not a comprehensive measure of the model's performance. Other important metrics, such as precision, recall, F1 Score, and Confusion matrix, provide a more complete assessment of the model's accuracy and effectiveness.

Precision, recall, F1 score, and confusion metrics are commonly used evaluation metrics in machine learning, particularly for classification tasks. Precision measures the accuracy of positive predictions, indicating the proportion of correctly predicted positive instances among all instances predicted as positive. Conversely, Recall assesses the model's ability to capture all relevant positive instances, representing the proportion of correctly predicted positive instances among all actual positive instances. F1 score is the harmonic mean of precision and recall, providing a balanced measure that considers both false positives and false negatives. Confusion metrics, often presented in a confusion matrix, detail the distribution of true positive, true negative, false positive, and false negative predictions, offering a comprehensive view of a model's performance (Sokolova & Lapalme, 2009).

Different measurements are used to assess the model performance, such as F1-Score, precision, and recall. It is important to understand how the model performs. The evaluation metrics show a remarkable average of 99% across all predicted classes, corresponding to the digits 0–9 in the MNIST dataset. One of the most important

evaluation metrics is F1-Score, which implies a strong balance of precision and recall, demonstrating the model's ability to reliably predict handwritten digits.

Furthermore, the dataset was separated into two sets, validation and training, with 18,000 images designated for validation and the rest for training. This allocation in validation ensures a representative sample across all classes of the handwritten digit images, and it is known as the Support. It contributes to a thorough evaluation of the model's performance over several digit classes. Moreover, another important classification evaluation metric is named the confusion matrix, which comes in the form of a matrix. It is a tabular representation that assesses the performance of a classification algorithm by splitting the classes into the predicted and actual classes of a dataset. Moreover, machine learning classification models commonly use it to evaluate the model's accuracy, precision, recall, and other performance metrics. It consists of four main components: true positives (correctly predicted positive instances), true negatives (correctly predicted negative instances), false positives (incorrectly predicted as positive instances), and false negatives (incorrectly predicted as negative instances). The values in the matrix aid in measuring the model's overall effectiveness and identifying areas for improvement (Mishra, 2019).

In order to show the performance of the model more straightforwardly, a confusion matrix is made for the model performance. It can be seen clearly in Figure 4 that each class has the true positives with an amount of 0.99, which means it has truly predicted the number on the image with a smaller number of false positives with no more than 0.01.

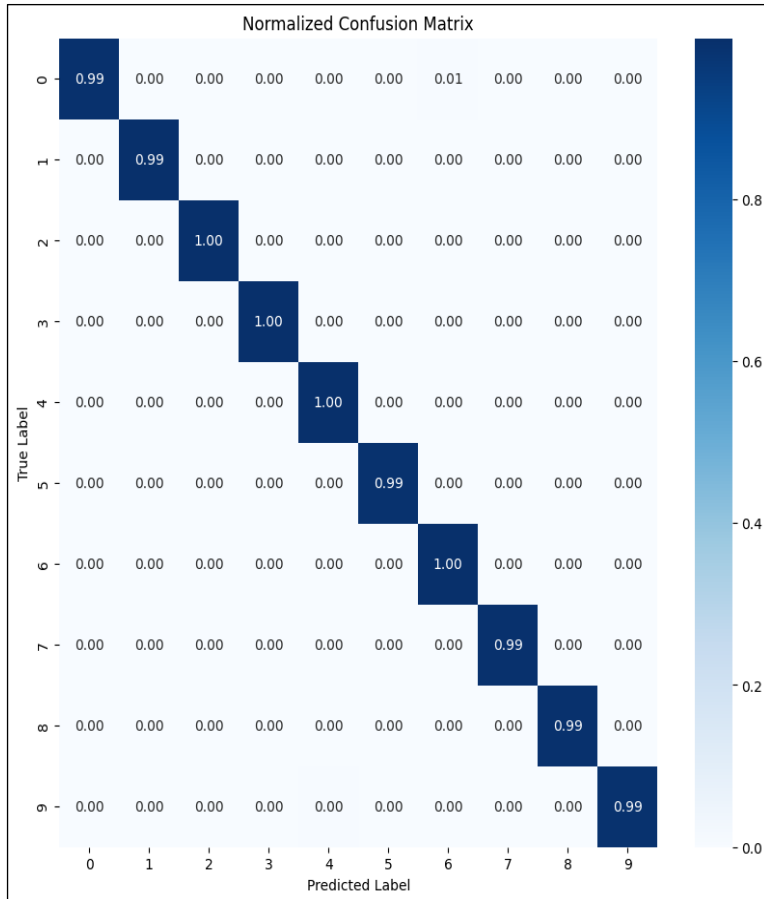


Figure (4): CNN model confusion matrix.

4.3 Steganography Encryption and Decryption

After the CNN model shows a high accuracy rate across all evaluation metrics, it indicates that we can proceed to the next stage, focusing on security. This involves the steganography process for extracting hidden content.

By converting the text messages into binary and applying the same process to the image, all the pixels are correctly aligned with the image pixels. This successful alignment indicates a successful steganography process, as confirmed when the text is accurately extracted from the predicted image by the CNN model. Consequently,

the original message is retrieved using all the equations applied in the encryption and decryption processes.

4.4 Discussion

The aim of this study is to provide strong, secure environment techniques that can securely send messages over the internet. A complex methodology, including cryptography, steganography, and CNN, is used; which are proposed and tested to achieve the goal of secure communication. This section evaluates the result, describing the implications of this study and suggestions for future researchers.

One of the main components of this proposed security technique is cryptography, in which plain text is encrypted into an unreadable form that unauthorized individuals cannot read. This technique decreases the possibility of revealing the original text and decrypting data illegally, and as such, it ensures that the data remains private during network transmission.

The steganography and cryptography are used together to hide the encrypted message within the digital image. The use of these two techniques enables discreet communications for sending text messages that are hidden in images and from outside observers. In the meantime, CNN processing is essential for predicting handwritten digits on images, embedding the encrypted message within the digital image containing the handwritten digit. Furthermore, this study demonstrates the strength and efficiency of the proposed method by presenting an effective text-hiding mechanism and successfully extracting encrypted messages from digital images through testing analysis.

Our study findings are compared with previous studies on secure networks for sending messages. Although some previous research has mainly concentrated on one part of the proposed methodology, few of them used a combination of the methodologies with advanced machine learning or deep learning, such as CNN. By combining these three different techniques, publications suggest different proposed rigorous solutions to problems of secure messaging in digital communication channels.

Our findings emphasize the significance of the proposed method, highlighting the importance of evaluating and validating security techniques for sending messages

over network channels. The primary evaluation of the findings focusses on accuracy of image prediction and the successful extraction of the original text message from the image. Further to these, the findings provide strong evidence for securely ensuring message delivery.

The primary challenge this research faces is the use of the MNIST dataset, which is limited to handwritten digit recognition and may not sufficiently represent the diversity of real-world images in practical communication scenarios. Hence, whether the proposed methods can be applied to more complex and diverse data is still an open question.

Future studies can focus more on improving the use of the CNN model by using transfer learning and adversarial training mechanisms, which are branches of the CNN model. Furthermore, the scalability and generalizability of the proposed methodology can be implemented and tested on several datasets.

The field of image steganalysis deals with multiple issues when using DL techniques. Limited training data restricts model diversity and effectiveness, while DL models struggle to detect unfamiliar steganographic methods, highlighting the need for better generalization. Malicious assaults make detection more challenging by misleading models, necessitating more robust defenses. Moreover, the absence of transparency in deep learning models diminishes confidence and comprehension in their outcomes. Addressing these issues requires developing more generalizable, resilient, and interpretable DL models tailored for steganalysis (Farooq & Selwal, 2023).

Steganalysis techniques, such as signature-based and statistical methods, are widely utilized to detect steganography. On the other hand, they often depend on prior experience of the particular steganographic algorithm employed for embedding the hidden message (De et al., 2024).

5. Conclusion

This research study introduces a novel strategy for improving the security of messages before sending them via networks. It used a technique that combined CNN for image classification with cryptography and steganography techniques for hiding data. It

displays high prediction accuracy and a durable mechanism for concealing text within images.

The proposed CNN model's performance shows 99% accuracy in recognizing handwritten digit numbers in the MNIST dataset, which means it is reliable and applicable in security applications. The precision, recall, and F1 score measurements provide a more detailed evaluation and highlight the model's effectiveness across multiple classes. Further to these, the used dataset elevates the proposed method's status as a benchmark for future research in the connection of neural networks and security applications.

This new approach illustrates a clear understanding of the technology and procedures used to provide a secure environment. The main evaluation of using hyperparameters, the explanation of selecting optimization techniques, and the strategic use of dropout layers all explain a scholarly approach to model design.

The proposed method is feasible for further security and investigation of neural network applications. Its successful combination of image classification using the CNN model and steganography for embedding technique lays the path for greater privacy and secrecy in communication systems for data exchange. The conclusions of this research study pave the way for conducting future research into hyperparameter optimizations, different dataset exploration, and the application of this technology to real-world issues in terms of security. Moreover, this study significantly contributes to the area and provides a stable platform for future advances in secure communication systems.

References:

- Alanzay, M., Alomrani, R., Alqarni, B., & Almutairi, S. (2023). Image Steganography Using LSB and Hybrid Encryption Algorithms. *Applied Sciences*, 13(21), 11771. <https://doi.org/10.3390/app132111771>
- Begum, M., & Uddin, M. S. (2020). Digital Image Watermarking Techniques: A Review. *Information*, 11(2), 110. <https://doi.org/10.3390/info11020110>
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. <https://link.springer.com/book/9780387310732>

- Bisong, E. (2019). Building Machine Learning and Deep Learning Models on Google Cloud Platform. (pp. 59-64). Berkeley, CA: Apress.
- Chandra, S., Paira, S., Alam, S.S., & Sanyal, G. (2014). A comparative survey of Symmetric and Asymmetric Key Cryptography. *2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE)*, 83-93. <https://doi.org/10.1109/ICECCE.2014.7086640>
- De La Croix, N. J., Ahmad, T., & Han, F. (2024). Comprehensive survey on image steganalysis using deep learning. *Array*, 100353. <https://doi.org/10.1016/j.array.2024.100353>
- Divyasree, S., Harichandana, N., Khushi, M., & Afreen, S. (2024). A Comprehensive Review on Advancements and Applications of Steganography. <https://doi.org/10.13140/RG.2.2.13568.44807>
- Dooley, J. F. (2018). History of Cryptography and Cryptanalysis. In *History of computing*. Springer, Cham. <https://doi.org/10.1007/978-3-319-90443-6>
- Dzhanashia, K., & Evsutin, O. (2024). Neural networks-based data hiding in digital images: Overview. *Neurocomputing*, 581, 127499. <https://doi.org/10.1016/j.neucom.2024.127499>
- Farooq, N., & Selwal, A. (2023). Image steganalysis using deep learning: a systematic review and open research challenges. *Journal of Ambient Intelligence and Humanized Computing*, 14(6), 7761-7793. <https://doi.org/10.1007/s12652-023-04591-z>
- Hossain, M. I., Kadir, S., Fagun, F. I., Samiul, I., Saukhin, R. Z., & Hossain, M. I. (2024, October). Enhanced CNN Approaches for Multi-Image Embedding in Image Steganography. In *2024 2nd International Conference on Information and Communication Technology (ICICT)* (pp. 100-104). IEEE. <https://doi.org/10.1109/ICICT64387.2024.10839645>
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*. <https://proceedings.mlr.press/v37/ioffe15.pdf>
- Karaçam, C., Algül, F. N., & Tavitt, D. (2021). Transmission of Time and Position Variable Cryptology in Fibonacci and Lucas Number Series with Music. *Journal of Mathematical Sciences and Modelling*, 4(1), 38-50. <https://doi.org/10.33187/jmsm.885876>
- Kich, I., El Bachir Ameer, Y. T., & Benhfid, A. (2020). Image Steganography by Deep CNN Auto-Encoder Networks. *International Journal*, 9(4)
- Kingma, D. P., & Ba, J. (2014). Adam: A method for Stochastic Optimization. arXiv preprint. <https://arxiv.org/abs/1412.6980>

- Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Li, L., Zhang, X., Chen, K., Feng, G., Wu, D., & Zhang, W. (2024). Image Steganography and Style Transformation Based on Generative Adversarial Network. *Mathematics*, 12(4), 615. <https://doi.org/10.3390/math12040615>
- Li, Q., Wang, X., Wang, X., Ma, B., Wang, C., Xian, Y., & Shi, Y.Q. (2020). A Novel Grayscale Image Steganography Scheme Based on Chaos Encryption and Generative Adversarial Networks. *IEEE Access*, 8, 168166-168176.
- Li, Y., & Liu, Q. (2021). A Comprehensive Review Study of Cyber-Attacks and Cyber Security; Emerging Trends and Recent Developments. *Energy Reports*, 7, 8176-8186. <https://doi.org/10.1016/j.egyr.2021.08.126>
- Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12), 6999-7019.
- Long, B., Chen, Z., Liu, T., Wu, X., He, C., & Wang, L. (2024). A Novel Medical Image Encryption Scheme Based on Deep Learning Feature Encoding and Decoding. *IEEE Access*, 12, 38382–38398. <https://doi.org/10.1109/ACCESS.2024.3371888>
- Majeed, M. A., Sulaiman, R., Shukur, Z., & Hasan, M. K. (2021). A Review on Text Steganography Techniques. *Mathematics*, 9(21), 2829. <https://doi.org/10.3390/math9212829>
- Mandal, B., Chandra, S., Alam, S. S., & Patra, S. S. (2014). A Comparative and Analytical Study on Symmetric Key Cryptography. *2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE)*, 131-136. <https://doi.org/10.1109/icecce.2014.7086646>
- Mishra, A. (2019). Evaluating Machine Learning Models. In *Machine Learning in the AWS Cloud*, A. Mishra (Ed.). <https://doi.org/10.1002/9781119556749.ch5>
- Priya, S., Abirami, S. P., Arunkumar, B., & Mishachandar, B. (2024). Super-Resolution Deep Neural Network (SRDNN) Based Multi-Image Steganography for Highly Secured Lossless Image Transmission. *Scientific Reports*, 14(1), 6104.
- Rana, M. S., Kabir, M. H., & Sobur, A. (2023). Comparison of the Error Rates of MNIST Datasets Using Different Type of Machine Learning Model. *North American Academic Research*, 6(5), 173-181. <https://doi.org/10.5281/zenodo.8010602>
- Sanjalawe, Y., Al-E'mari, S., Fraihat, S., Abualhaj, M., & Alzubi, E. (2025). A deep learning-driven multi-layered steganographic approach for enhanced data security. *Scientific Reports*, 15(1), 4761. <https://doi.org/10.1038/s41598-025-89189-5>

Sokolova, M., & Lapalme, G. (2009). A Systematic Analysis of Performance Measures for Classification Tasks. *Information processing & management*, 45(4), 427-437.

Sony, S., Dunphy, K., Sadhu, A., & Capretz, M.A. (2021). A Systematic Review of Convolutional Neural Network-Based Structural Condition Assessment Techniques. *Engineering Structures*, 226, 111347.

<https://doi.org/10.1016/j.engstruct.2020.111347>

Subramanian, N., Elharrouss, O., Al-Maadeed, S.A., & Bouridane, A. (2021). Image Steganography: A Review of the Recent Advances. *IEEE Access*, 9, 23409-23423.

Tomasi, C., & Manduchi, R. (1998). Bilateral Filtering for Gray and Color Images. *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 839-846.

<https://doi.org/10.1109/iccv.1998.710815>

Yu, J., Zhang, X., Xu, Y., & Zhang, J. (2023). CRoSS: Diffusion Model Makes Controllable, Robust and Secure Image Steganography. 37th Conference on Neural Information Processing Systems (NeurIPS 2023). *ArXiv, abs/2305.16936*.

<https://doi.org/10.48550/arxiv.2305.16936>

ئاسایشی ئالوگۆرى زانىارىيەكان: شىوازيكى يەكگرتوو بە بەكارھيئەتلىرى تۆرە دەمارىيە پىچاوپىچەكان وستىگانۆگرافى وئىنە دىجىتالى (MNIST)

پوختە:

لە ئىستاندا پەيوەندىيە دىجىتالىيەكان بەشىۋەيەكى بەربلاۋى بەكاردەھيئەتلىرى بۆيە پاراستىنى زانىارىيەكان لەكاتى گواستەنەۋەدا بوۋەتە پىۋىستىيەكى ھەنوكەي. ئەم توۋىزىنەۋەيە چەمكىكى نوۋ دەخاتە پوو لە رېگەي بەكاھىنەنى دوو مىسۇدو، ستىگانۆگرافى (Steganography) و فېربوۋنى قوۋل (Deep Learning). لەم كارەدا ژمارە دەستىۋوسەكانى كۆمەلە داتاكانى پەيمانگاي نىشتەمانى ستاندارد و تەكنەلۇژىيە گۇراۋ (MNIST) بەكاردەھيئەتلىرى بۇ دروستكرىنى سىستەمىك كە وئىنە بەكاردەھيئەتلىرى بۇ شاردنەۋەيە پەيامە نەئىيەكان. تەكنەلۇژىيە شاردنەۋەيە زانىارى لەگەل تۇرپىكى دەمارى پىچاوپىچە (CNN) تېكەل كراۋە، كە دوو كار ئەنجام دەدات: يەكەمىيان شاردنەۋەيە زانىارىيە و دوۋەمىيان دۇزىنەۋە و پىشېبىنىكرىنى پەيامە شاراۋەكانە بە شىۋەيەكى سەرسۈرپەنەر كە دەگاتە 99%. ئەم رېگەيە ئاسايشى گواستەنەۋەيە زانىارىيەكان مەسۇگەر دەكات ۋە دلىيەي دەدات لەۋەي كە

تهنئا ٺه وه سانهى كلىلى دروستيان بۆ كردنه وه يه كى كۆده كان هه يه ده ستيان به زانياريه نهينيه كان بگات. تىكه لكردنى ئهم دوو رېگهيه پىكه وه ستانداردىكى نوئ بۆ ئالوگۆرى زانياريه نهينيه كان له تۆره كانى ئىنته رنئىدا دههينيه كايه وه، ههروه ها چوارچيوه يه كى بههيز داين دهكات بۆ كه مكرده وه لاوازى سيسته ميه كانى پهيوه ندييه دى كه له ئىستادا به دى ئه كرىت. بهم شيوه يه، دۆزينه وهى زانياره شاره وه كان زۆر ئه ستهم ده بىت.

آمن تبادل البيانات: طريقة متكاملة باستخدام الشبكات العصبية الالتفافية وإخفاء المعلومات في الصور الرقمية MNIST

الملخص:

مع الانتشار و الاستخدام الواسع للاتصالات الرقمية ، أصبح نقل المعلومات بطريقة آمنة ضرورة ملحة. تقدم هذه الورقة مفهوماً جديداً يجمع بين تقنية إخفاء البيانات (Steganography) والتعلم العميق (Deep Learning) لحماية البيانات السرية أثناء التنقل. يستخدم العمل المقترح الأرقام المكتوبة بخط اليد لمجموعة بيانات المعهد الوطني المعدل للمعايير والتكنولوجيا (MNIST) لإنشاء نظام يستخدم الصور لإخفاء الرسائل السرية. ويتم دمج تقنية إخفاء البيانات مع الشبكة العصبية الالتفافية (CNN)، وهي مصممة للعمل بشكل مزدوج والغرض منها إخفاء البيانات وتوليد عمليات الكشف وتنبأ بوجود رسائل مخفية بدقة مذهلة تبلغ 99٪. هذه الطريقة تضمن وصول فقط الأشخاص الذين لديهم مفتاح فك التشفير الصحيح إلى المعلومات السرية وبالتالي ضمان أمنية تنقل البيانات. دمج هاتين الطريقتين معا يصنع معياراً جديداً لتبادل آمن للبيانات في الشبكات الحديثة، مما يوفر إطاراً قوياً يخفف من نقاط الضعف في أنظمة الاتصالات الحالية. وهكذا، فإن الكشف عن التضييل في الصور يعد دقيقاً للغاية ويساهم بشكل كبير في تطوير حماية البيانات في مواجهة التهديدات الاصطناعية المستجدة.