Journal of Advanced Zoology
ISSN: 0253-7214
Volume 45 Issue 3 Year 2024 Page 602-622

Comparative Performance Analysis Of Deep Learning-Based Image Steganography Using U-Net, V-Net, And U-Net++ Encoders

Sapna Kaneria[1*], Dr. Varsha Jotwani[2]

[1]Ph.D Scholar, Department of Computer Science, RNTU Bhopal*
[2]Professor and HOD, Department of Computer Science & IT, RNTU Bhopal

***Corresponding Author:** Sapna Kaneria*
[*]Ph.D Scholar, Department of Computer Science, RNTU Bhopal

Abstract

Digital Imaging steganography is the act of hiding information in a cover picture in a way that can't be found or recovered. Three main types of methods are used in digital image steganography: neural network methods, spatial methods, and transform methods. The pixel values of an image are changed by spatial methods to embed information. On the other hand, the frequency of the image is changed by transform methods to embed information that is hidden. There are methods that use neural networks to hide things, and this is what the suggested method is all about. Through digital image steganography, this study looks into how deep convolutional neural networks (CNNs) can be used. With the increasing concerns about data infringement during transmission and storage, image steganography techniques have gained attention for hiding secret information within cover images. Traditional methods suffer from limitations such as low embedding capacity and poor reconstruction quality. To address these challenges, deep learning-based approaches have been proposed in the literature. Among them, the Convolutional Neural Network (CNN) based U-Net encoder has been extensively studied. However, its comparative performance with other CNN-based encoders like V-Net and U-Net++ remains unexplored in the context of image steganography.

In this paper, we implement V-Net and U-Net++ encoders for image steganography and conduct a comprehensive performance assessment alongside U-Net architecture. These architectures are utilized to conceal a secret image within a cover image, and a unified and robust decoder is designed to extract the hidden information. Through experimental evaluations, we compare the embedding capacity, stego quality, and reconstruction quality of the three architectures. The U-Net architecture outperforms V-Net and U-Net++ in terms of embedding capacity and the quality of stego and reconstructed secret images. This research provides valuable insights into the effectiveness of different deep learning-based encoders for image steganography applications, aiding in the selection of appropriate architectures for securing digital images against unauthorized access.

keywords- U-Net, V-Net, U-Net++,deep leaning ,CNN, image steganography.

I INTRODUCTION

Secret information hiding is the science and art of keeping private information hidden so that only the person who is meant to get it can find it [1, 2]. For example, steganography is both an art and a science. As the name suggests, a cover object is something that everyone can see but that holds private information. Numerous types of cover objects can be used in various steganographic methods [3–9]. Texts, videos, audio files, and still images are all examples of cover items. Photographs are thought to be good cover things for steganography. They have a lot of duplicate pixel values, and our eyes aren't as good at picking out small features. It is called the cover image or the host image when the original picture doesn't have the secret message on it, and it is called the stego image when it does. Evaluating a picture steganography method usually involves looking at five different areas: perceptual transparency (visual quality), payload (embedding) capacity, security, temperature resistance, and computational costs [10, 11]. Visual quality is good when the human vision system (HVS) can't tell the difference between the cover picture and the stego image. What sets the embedding capacity, which is measured in bits per pixel (bpp), is how much private information can be hidden in a cover image. The ability to add more secret data to the host picture is enabled by a larger payload amount. Being able to keep private information safe even if someone who isn't supposed to see it finds it in the stego picture is what we mean by "security." It is possible to get both better payload and imperceptibility by using picture steganography.

Thieves who are trying to steal information will not notice a steganographic scheme with a less deformed stego picture as easily as one with a highly deformed stego picture. Perfect steganography requires a system that can really hide things, and the hidden items must look better than anything else on the market. Although visual quality is related to embedding ability, the relationship is the opposite way around. A big drop in the other number happens when one parameter is raised. Finding a balance between the two factors based on the user's needs, which can be different from one app to the next [12–14], is the only way to solve this problem. Space-based image steganography and transform (or frequency-based) domain-based image steganography are the two main types of picture steganography methods. Something interesting about this. Secret messages are directly hidden in the cover file's pixels when spatial image steganography is used. LSB (Least Significant Bit) substitution methods, PVD (Pixel Value Differencing) methods, QIM (Quantization Index Modulation) methods, and LSB matching methods are the major types of methods used to hide data in the spatial domain. Our class uses LSB substitution more than any other way. In order to hide secret data in a cover image using this method [15–17], the k least significant bits (LSBs) of the pixel value of the cover picture are switched directly with the k bits of the secret message. A stego picture is made as a result. It is possible for PVD-based algorithms to figure out how many bits should be in a cover picture [18] by calculating the differences between blocks of pixels. When using the LSB matching method, the bits of the secret message are compared to the least significant bits (LSBs) of the pixels and cover picture. Randomly adding or taking away one from the pixel value of the cover picture [19] if there is no match. Following these steps is pretty easy, and they don't use much computer power. Despite this, the stego image gets worse as the number of embedded bpp grows, and it is easier for frequency-based attacks like filtering and compression to damage it. For transform domain based image steganography to work, messages that need to be kept secret are hidden in the cover image's transform coefficients. The problems with spatial domain methods like not being able to be recovered or seen can be fixed with data hiding strategies that use the transform domain. This process changes the host picture into the frequency domain by finding its frequency coefficients. They are then shown. To change the picture back from the frequency domain to the spatial domain, an inverse transform is also used. Next, these factors are embedded. A lot of research is done on different transform domain techniques, such as discrete Fourier transforms (DFT) methods, discrete cosine transforms (DCT) methods, discrete wavelet transforms (DWT) methods, and integer wavelet transforms (IWT) methods! [8, 13, 20]

Image Steganography

Images are great for hiding information for a number of reasons. Some of the many things that make up an image are its bit depth, colors, borders, corners, measurements, and metadata [6]. This is the most important reason for this result. Because of these factors, it is relatively easy to hide a payload inside a picture. Along with the properties mentioned above, the metadata of some picture formats can also be changed to add information. Graphics Interchange Format (GIF) image files have a pallet that has all the colors that were used to make the picture. The image looks the same, but by changing its color map, it hides some details [4]. This is because the picture has been changed. Almost all images don't change when the numbers of the pixels do, which brings us to the second reason. To hide parts of a picture inside a cover image, many steganography methods involve changing the pixels that aren't important. Most pictures are split into three color channels that are each eight bits wide: red, green, and blue. This leaves a lot of room for information to be stored. Image steganography is based on a basic framework that can be seen in Figure 1.1. The person sending the message puts together this steganographic message by using two parts: the message and the payload. It is the process of encoding that makes up the steganographic message. In order to make things clearer, we will talk about a number of different encoding methods in the parts that follow. Right after the encoding process is done, the steganographic picture is sent to the person who is supposed to get it. The steganographic picture can be sent over either secured or unprotected channels because people who get access to it won't be able to use it to their advantage unless they are aware that it has a hidden payload. The person who receives the steganographic picture will use a process similar to decoding to figure out what the message is. However, the decoding process is the exact same as the encoding process used in most steganographic methods; it is just done backwards. Once the decoding process is done, the receiver will be able to see the hidden picture.

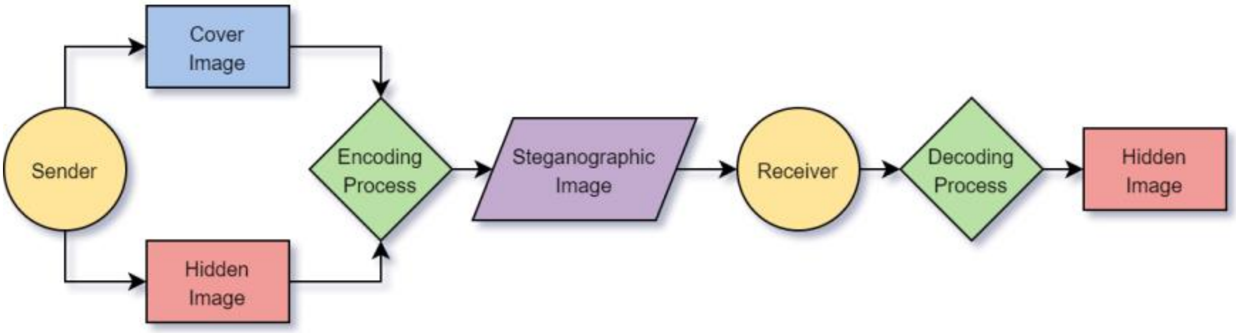


Figure 1 Image Steganography

II RELATED WORKS

Baluja et al. (2019) [21] The first showed an end-to-end DS model made of convolution networks that could hide an RGB picture in another RGB image of the same size. The hiding capacity of this model was 24 bpp, which was higher than the concealment capacity of previous methods, which were less than 0.5 bpp. They also kept the difference between the container image and the cover image as small as possible while reducing the difference between the reconstructed image and the secret picture. Because their method had less strict rules, the secret information didn't have to be revealed in a certain way.

Zhu et al. (2018) [22] HiDDeN is a network for hiding binary information in photos that was made to make the model more reliable for both steganography and picture watermarking. Noise layers had to be added to this network. By adding a steganalysis network as a threat, HiDDeN was also able to improve its ability to stop steganography. It was hard for the network to deal with new sounds, though, because the rebuilt binary information still had a high two-bit error rate and there were only a few types of noise layers it could use.

Luo et al. (2020) [23] Instead of embedding binary messages directly, message coding was used. This made the information that was retrieved more reliable by lowering the number of bit errors. Also, to get robustness without modeling the distortion, they used artificial neural networks to cause distortion through adversarial networks.

Qin et al. (2020) [24] Coverless steganography was done using a method that suggested CNNs and GANs be used. With the help of adversarial example approaches, Shang et al. [25] were able to make the algorithm safer and offer a DS strategy. It was suggested by Zhu et al. [26] that an image-hiding convolution neural network could be used. This network would use a residue network, pixel shuffle, and picture encryption. Wang et al. [27] were the first to use Transformer for picture steganography, and they were able to improve the quality of the images they used. Inspired by examples of hostile events that can happen to anyone.

Zhang et al.(2020) [28] The goal of this study was to come up with a universal deep hiding model (UDH) and look into how to make cover-independent changes that would hide a hidden picture in a number of unknown cover images. It is possible to change UDH to protect the rights to recorded movies. Steganography can be used on digital pictures, but printed photos and digitally projected photos can also be used to hide digital data that can't be seen. When it comes to printing and photos, both deep photographic steganography [17] and light field messaging (LFM) [19] are good ways to fix problems with pictures. To make the model more reliable,

Chen et al. (2021) [29] this person came up with a low-frequency picture DS method. Based on the study that Yin and his colleagues did, they suggested an image DS method that has a unique fine-tuning network structure. By fixing the problem caused by the stego matrix being rounded, which led to a loss of accuracy, this method worked. Pan and his coworkers came up with a picture DS method based on deep reinforcement learning. With this method, secrets could be kept hidden in the area in a way that worked with the environment. This is what Wang et al. [30] showed when they were working on steganography: a DS system built on a capsule network that can send extra data.

Akshay Kumara et.al. (2023) [31] these days, data security is very important because more and more things are done online. Different types of transmission channels are now commonly used to send data and information from one body to another. Data and information are also stored on virtual online repositories. A lot of the information and data that is being sent is thought to be very private. Because of this constant flow of technology advances, many data security methods have been created to keep data safe while it is being sent or stored. These include encryption, data hiding, and others. It is suggested in this study that one image can be hidden inside another image. For feature extraction and analysis, CNN is used as part of the method. Following the rules of the auto-encoder design, the model that has been submitted works. Two parts make it up: the Hiding network/Encoder part processes the secret image and pulls out n features; the other part adds those features to the cover image in a way that keeps the cover image's original look while hiding the secret image. That's the other part of this network that pulls out the hidden secret picture and then makes it again in its original form. This paper is the Reveal network/Decoder. To test the suggested model, we use the ImageNet and Pascal-VOC datasets, which have pictures of different sizes. They can test the model's ability to hide the picture and find it again by using PSNR and SSIM as performance metrics. CNN designs have become more popular because they can extract features automatically, have a smaller feature map, operate very accurately, and be used in many different areas [32]. Common uses for CNN systems include pattern recognition [33], classification [34], object recognition [35], and image segmentation [36]. According to the newest study, these networks can also be used for steganography. Numerous people are familiar with the U-Net, V-Net, and U-Net++ designs that are used for segmenting images. Researchers are looking into image steganography, which makes these methods more useful.U-Net is a convolutional neural network architecture that has gained popularity, especially in image segmentation tasks. The architecture is characterized by its U-shaped design, consisting of a contracting path and an expanding path. Here's a detailed description of the U-Net architecture.

Sapna Kaneria, Dr. Varsha Jotwani (2024) [41] covers the study about Advancements in Digital Steganography. This technique stands in contrast to encryption, which relies on mathematical algorithms to encode information, making it indecipherable without the appropriate decryption key. While both approaches serve the overarching goal of information security, they differ fundamentally in their methodologies. In the realm of steganography, researchers have explored various techniques and methodologies, aiming to enhance the effectiveness and security of data concealment. These techniques encompass both linguistic and technical steganography, each offering unique advantages and challenges in concealing information within different types of data. the literature review highlights the differentiation within watermarking techniques, which can be categorized as robust or fragile watermarking. Robust watermarking focuses on embedding information that can withstand various manipulations or attacks, while fragile watermarking is designed to be highly sensitive to any alterations, making it suitable for applications where data integrity verification is crucial. In the modern era, where vast amounts of data are transmitted across various digital channels, the need for confidentiality and data integrity is paramount. While encryption plays a vital role in securing information by encoding it with complex algorithms, steganography offers an alternative approach. Instead of making data indecipherable, steganography focuses on hiding data within seemingly innocuous carriers, whether they are images, audio files, or other forms of media. The significance of steganography lies in its ability to conceal data in plain sight, making it exceedingly challenging for unauthorized individuals to even detect the presence of hidden information, let alone decipher it. This covert communication method can be especially valuable in situations where overt encryption might arouse suspicion or where subtle data exchange is necessary. A review paper on steganography would provide an in-depth analysis of the key concepts, techniques, applications, and recent advancements in the field of steganography. Steganography is the art and science of hiding information within other data in such a way that it remains undetectable to unintended recipients.

III DEEP LEARNING ARCHITECTURES

CNN designs have become more popular because they can extract features automatically, have a smaller feature map, operate very accurately, and be used in many different areas38. Common uses for CNN systems include pattern recognition39, classification40, object recognition41, and image segmentation42,43. According to the newest study, these networks can also be used for steganography9. Numerous people are familiar with the U-Net, V-Net, and U-Net++ designs that are used for segmenting images. Researchers are looking into image steganography, which makes these methods more useful.U-Net is a convolutional neural network architecture that has gained popularity, especially in image segmentation tasks. The architecture is characterized by its U-shaped design, consisting of a contracting path and an expanding path. Here's a detailed description of the U-Net architecture:

Convolutional Layer:

The output of a convolutional layer is computed using the convolution operation with learnable weights (parameters) represented by W and biases b. Let x be the input feature map, and y be the output feature map:

$y=\sigma(W*x+b)$ Eq.1

where $**$ denotes the convolution operation, σ is the activation function (usually ReLU), and ++ represents element-wise addition.

1. **Max-Pooling Layer:** Max-pooling down samples the input feature map by selecting the maximum value in each local region. Let x be the input feature map, y be the output feature map, and $\downarrow\downarrow$ denote the max-pooling operation $\downarrow y=x\downarrow$

2. **Deconvolution Layer (Transposed Convolution or Up sampling):**

The deconvolutional layer increases the spatial resolution of the input feature map. Let x be the input feature map, y be the output feature map, $\uparrow\uparrow$ denote the deconvolution operation, and W be the learnable weights:

$y=\sigma(W\uparrow x+b)$ Eq.2

Concatenation Layer (Skip Connection): The concatenation layer combines feature maps from the contracting path with the corresponding layers in the expanding path. Let 1x1 and 2x2 be the feature maps being concatenated:

$y=\mathrm{concat}(x1,x2)$ Eq.3

In the U-Net architecture, the contracting path involves a series of convolutional layers followed by max-pooling, while the expanding path involves deconvolutional layers. Skip connections connect corresponding layers between the contracting and expanding paths through concatenation.[37]

U-Net architecture

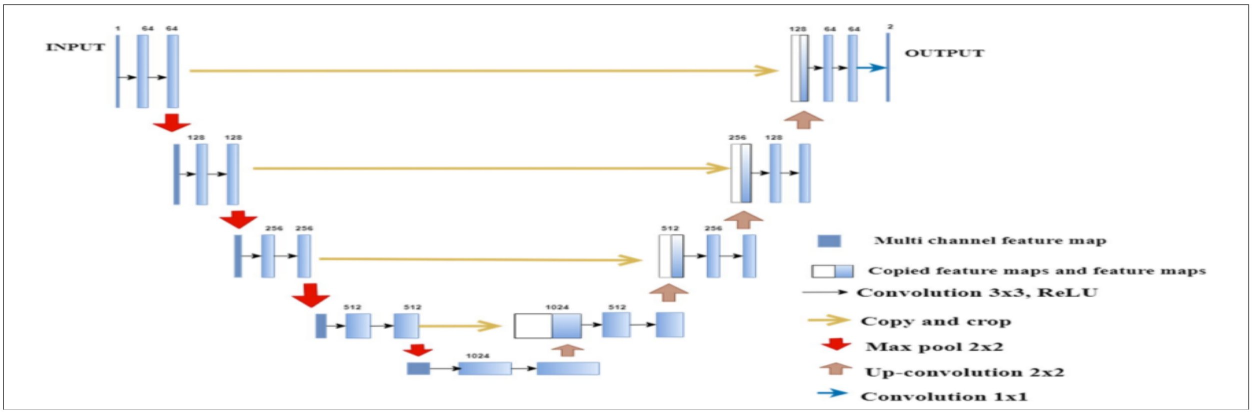


Figure 2. U-net architecture.

VNet Architecture

Input Layer: This layer takes 3D medical image data (e.g., MRI or CT scans) as input.

Encoder Layers: Convolutional layers with ReLU activation function: These layers perform feature extraction from the input data. Down sampling layers (often implemented as max-pooling): These layers reduce the spatial dimensions of the feature maps while increasing the depth, capturing hierarchical features.

Bridge: Convolutional layers with ReLU activation function: These layers further refine and consolidate features extracted from the encoder layers.

Decoder Layers: Up sampling layers (often implemented as transposed convolution or interpolation): This adds more space to the feature maps while decreasing their depth when these layers are used. Concatenation with corresponding encoder feature maps: Feature maps from the encoder are concatenated with decoder feature maps to provide high-resolution spatial information combined with rich contextual information. Convolutional layers with ReLU activation function: These layers further refine the features.

Output Layer: Convolutional layer with softmax activation function: This layer assigns a probability to each voxel (3D pixel) indicating the likelihood of belonging to a particular class (e.g., foreground/background). The functions involved in these layers include:

Convolution Operation: The convolution process includes two steps: sliding a kernel or filter over the input feature map and finding the dot product between the kernel weights and the attached input values. Mathematically, it can be represented as:

$Yi,j=\sum m,nXi+m,j+n\times Km,n+b$ Eq.4

Where:

- Yi,j is the output feature map at position (i,j).
- Xi+m,j+n is the input feature map.
- Km,n is the kernel/filter.
- b is the bias term.

ReLU Activation Function: To make sure the network isn't linear; the Rectified Linear Unit (ReLU) activation function is applied to the output of the convolutional cells one element at a time. It can be mathematically defined as:

$f(x)=\max(0,x)$ Eq.5

Max-Pooling Operation: Max-pooling is used to reduce the size of the feature maps' spatial dimensions. The highest number that can be kept in each pooling zone is what this method does. It helps capture the most important traits while minimizing the amount of computing complexity. It can be thought of mathematically as the process of getting the highest value possible in a pooling area.

Transposed Convolution Operation (Deconvolution): Transposed convolution is used in the decoder layers to upsample the feature maps. It is mathematically represented as an inverse convolution operation that spreads the information over a larger area.

Softmax Activation Function: Softmax activation is applied to the output layer to convert the raw scores into probabilities. It ensures that the sum of probabilities across all classes is equal to 1. Mathematically, it can be represented as:

$P(y=j|X)=\sum j'eXi,j'eXi,j$ Eq.6

Where:

- P(y=j|X) is the probability of class j given input X.
- Xi,j is the raw score for class j at position (i,j).
- e is the base of the natural logarithm.

These mathematical functions are applied iteratively through the layers of the V-Net architecture to extract and process features from the input medical images for accurate segmentation.

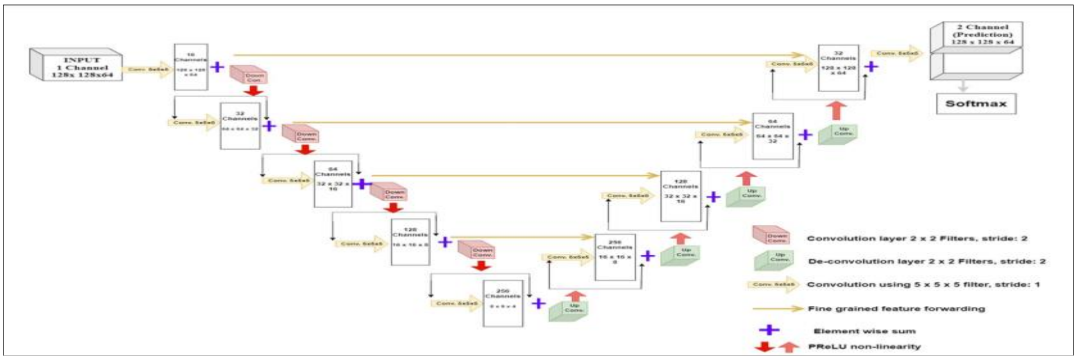


Figure 3. V-Net architecture.

The V-Net architecture is a deep learning model primarily designed for medical image segmentation tasks, particularly in the context of biomedical image analysis such as MRI or CT scans. It was introduced to address the challenges of segmenting 3D medical images efficiently and accurately.

U-Net++ Architecture

The U-Net++ architecture is an extension of the U-Net model, which incorporates nested skip connections to improve feature representation and segmentation performance. Let's describe the architecture of U-Net++ using mathematical functions:

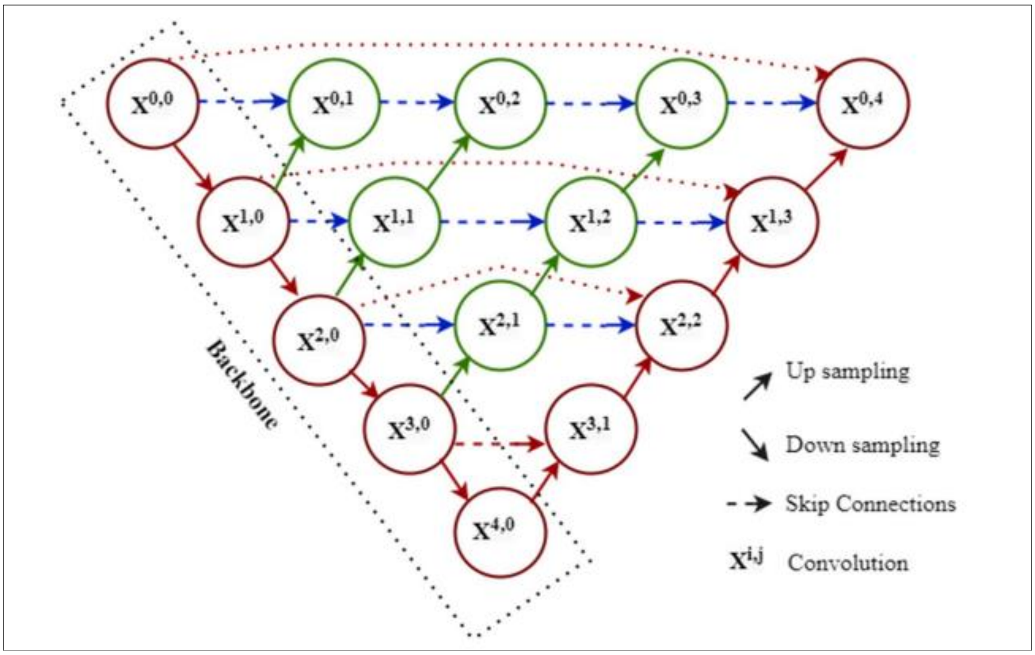


Figure 4 U-Net++ Architecture

Input Layer: The input layer represents the input image data, denoted as X.
Encoder Blocks: The encoder consists of multiple convolutional blocks followed by down sampling operations such as max-pooling or strided convolution.Let Convi represent the i-th convolutional block, where 1,2,...,
i=1,2,...,n. Each convolutional block involves convolutional layers with ReLU activation functions. The output feature maps of the encoder blocks are denoted as FiE, where i=1,2,...,n.
FiE=Convi(Fi−1E) Eq.7

Skip Connections (Nested Skip Connections): Skip connections are established between corresponding encoder and decoder blocks, as well as between encoder blocks at different resolutions. Let Skipij denote the skip connection from the j-th encoder block to the i-th decoder block. The skip connection combines the feature maps from the encoder block j with the feature maps from the decoder block i. The combined feature maps are represented as FiD.
FiD=Concat(Fi−1D,Skipij) Eq.8

Decoder Blocks: The decoder consists of multiple convolutional blocks followed by upsampling operations such as transposed convolution or interpolation. Similar to encoder blocks, let ConviD represent the i-th convolutional block in the decoder. The output feature maps of the decoder blocks are denoted as FiD, where i=n,n−1,...,1.
FiD=ConviD(Fi+1D) Eq.9

Output Layer: The output layer generates the segmentation mask or the reconstructed image. Let Output denote the output layer function.
Y=Output(F1D) Eq.10

IV PROPOSED SYSTEM

A Convolutional Neural Network (CNN) for steganography is tested using three different deep learning architectures: U-Net, V-Net, and U-Net++. Although the U-Net architecture has been tested for picture steganography before, similar CNN-based methods like V-Net and U-Net++ have not been compared to how well it works for that purpose. Therefore, we first test the V-Net and U-Net++ architectures in the setting of image steganography and then see how well they do compared to the U-Net architecture. Its main addition is a comparison of U-Net, V-Net, and U-Net++ architecture-based steganography, looking at different performance factors. The most important efforts are listed below.An image-in-image steganography method built on U-Net, V-Net, and U-Net++ has been created to protect privacy in data storage and communication. Deep learning was used to make an encoder that can decode stego images made by any of the three encoders that were presented. Structures that have been suggested hide a cover picture that is the same size as the hidden image, which is N×N. Engineers from U-Net, V-Net, and U-Net++ use different types of architectures to hide the secret image inside the cover image. The secret picture is also taken from the stego image made by any of the encoder architectures using a common decoder architecture. Before, different designs were used for the methods that were used.The goal of this work is to test how well image-in-image steganography methods that use deep learning work. Figure 4 shows a design that shows three different deep learning systems. The U-Net, V-Net, and U-Net++ architectures work as encoders to hide a secret picture behind a cover image. Also, we create a one-of-a-kind decoder architecture to get the secret image from the stego image that is hidden.

Encoder Architecture: the input image as X, which represents the cover image. The goal of the encoder is to hide a secret image S into the cover image, producing a stego image Y.For the encoder architecture, we can represent it as a function E that takes the cover image X and the secret image S as input and produces the stego image Y as output:
$Y=E(X,S)$ Eq.11
The encoder E can be implemented using various deep learning architectures such as U-Net, V-Net, or U-Net++. These architectures consist of multiple layers of convolutional, pooling, and activation functions.

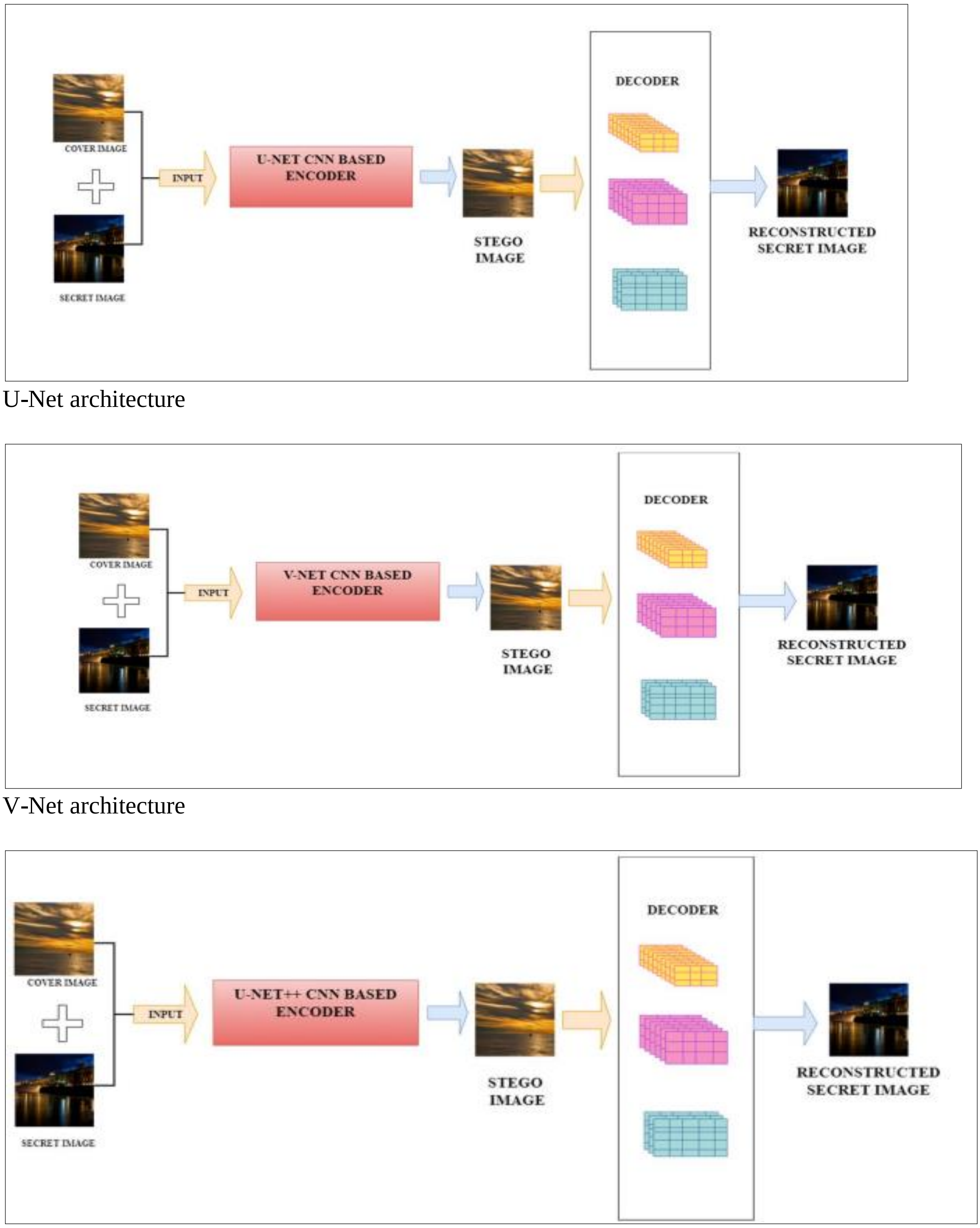
Decoder Architecture: The decoder architecture is responsible for extracting the hidden secret image S from the stego image Y. Let's denote the output of the encoder (i.e., the stego image) as Y′. Similar to the encoder, we can represent the decoder as a function D that takes the stego image ′Y′ as input and produces the extracted secret image ′S′ as output:
$S'=D(Y')$ Eq.12
The decoder D can also be implemented using deep learning architectures, typically involving convolutional layers followed by activation functions.

Convolutional Layers: Let CLi represent the i-th convolutional layer in the decoder, where =1,2,...,i=1,2,...,n. The output of each convolutional layer is denoted as Fi, which represents the feature maps produced by the layer.
$Fi=CLi(Fi-1)$ Eq.13

Concatenation of Convolutional Layers: After each convolutional layer, we perform concatenation to combine the feature maps from previous layers. Let's denote the concatenation operation as
$Fconcat=CAT(F1,F2,...,Fn)$ Eq.14

Output Layer: Finally, the output layer of the decoder produces the extracted secret image ′S′. Let's denote the output layer as Output.
$S'=Output(Fconcat)$ Eq.15

Architecture of Encoders: In our proposed steganography techniques, we utilize three distinct fully connected Convolutional Neural Network (CNN) architectures: U-Net, V-Net, and U-Net++. These architectures are responsible for generating a stego image that conceals the secret image within the cover image.[38]



U-Net architecture

V-Net architecture

U-Net++ architecture

Figure 5. Block diagram of the proposed steganography techniques (A) U-Net architecture based encoder; (B) V-Net architecture based encoder; (C) U-Net++ architecture based encoder.

Architecture of Decoder: They make a CNN-based decoder that is both unique and reliable as part of our study. The main goal of the decoder is to find hidden pictures in the stego images that are made by any encoder based on U-Net, V-Net, or U-Net++. The decoder has eleven convolution layers, shown in Figure 5. Each has a different kernel size, running from 3x3 to 4x4 to an impressive 5x5. The decoder network can capture features better because each kernel has many filters. Additionally, the convolution (CL) layers are in charge of creating feature maps. It is necessary to combine (CAT) several convolutional layers in order to get the most important meaning information from the feature maps. The model can learn better because of all of these factors. Within Table 2, it can see details about the output size and the number of factors in each layer.

"conv" means "convolution," and "level" shows the convolution layer's level. The first column shows information about the layers. The second column shows the output size that goes with each input layer. The number of factors that are present at each layer of the network is shown in the last column.

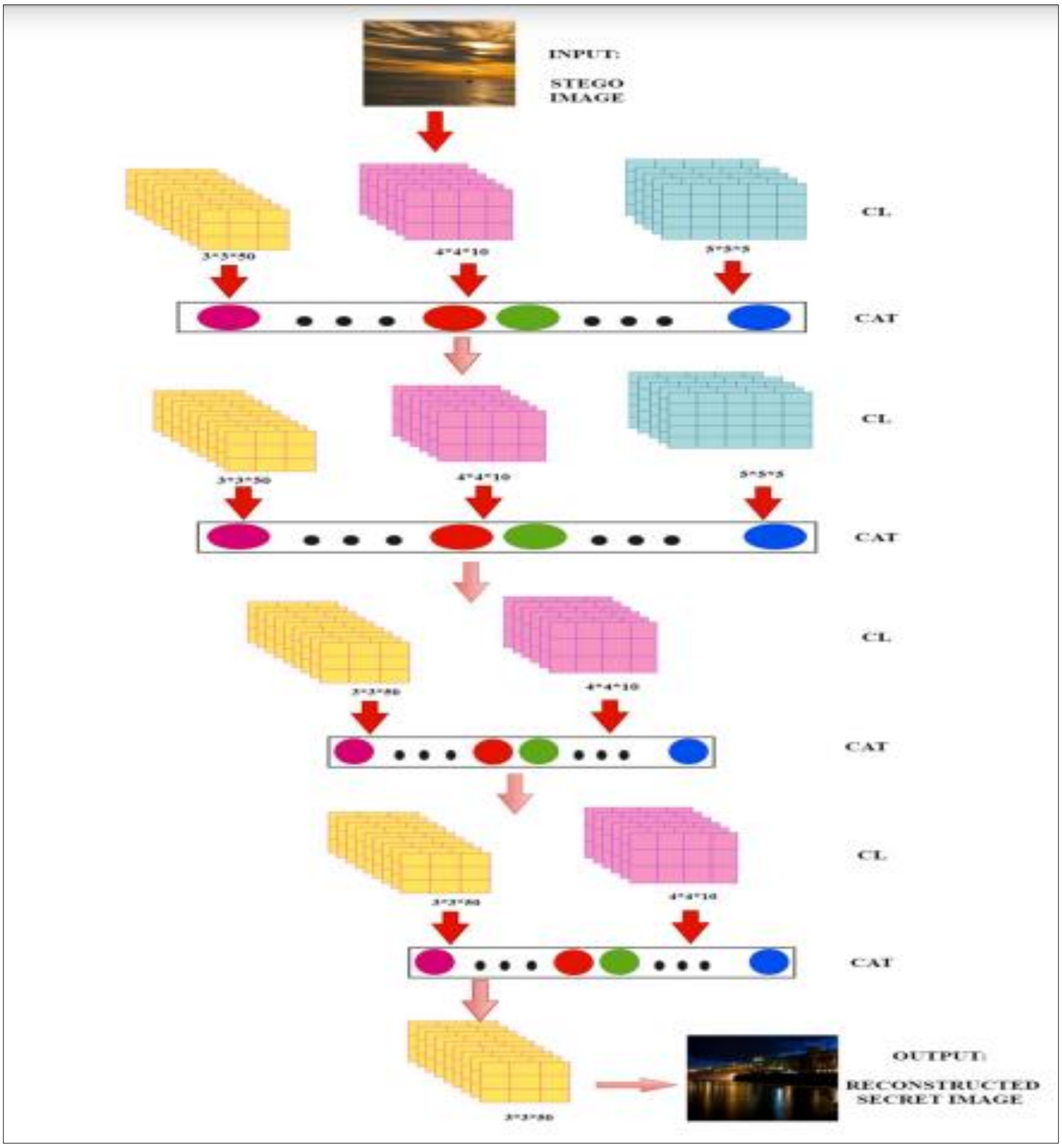


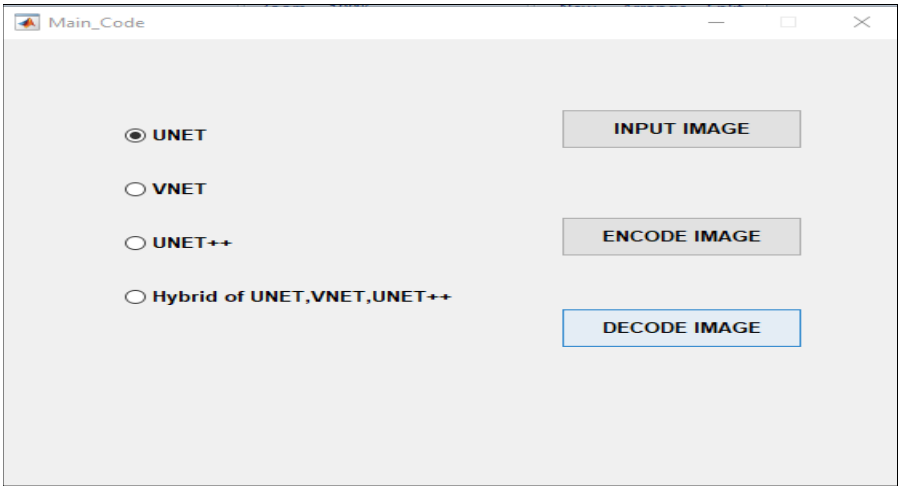
Figure 6. Decoder network architecture

V EXPERIMENTS EVOLUTION

The simulation results demonstrate the comparative performance of three deep learning-based encoders, U-Net, V-Net, and U-Net++, for image-in-image steganography. Through comprehensive evaluations, it is revealed that while all three architectures effectively conceal secret images within cover images, the U-Net architecture exhibits superior embedding capacity and produces stego images with higher quality and more accurately reconstructed secret images compared to V-Net and U-Net++

Encoding (Hiding): During encoding, the secret image is embedded into the cover image in such a way that it becomes imperceptible to the human eye. This process typically involves manipulating the pixel values of the cover image to accommodate the data from the secret image. Various techniques can be employed for this purpose, including least significant bit (LSB) substitution, which involves replacing the least significant bits of the cover image pixels with the bits of the secret image.

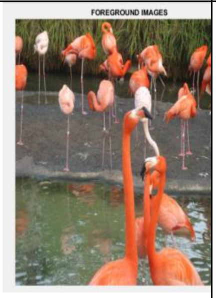
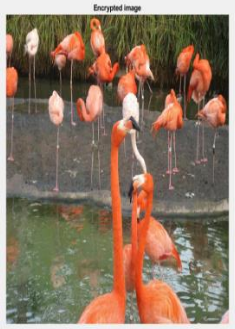
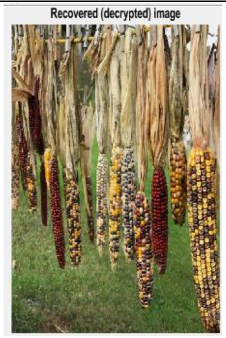
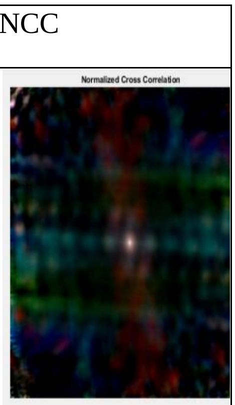
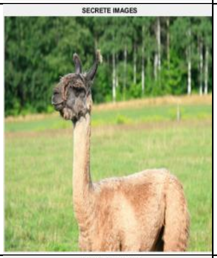
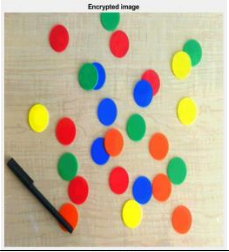
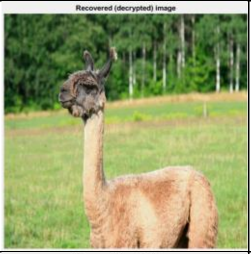
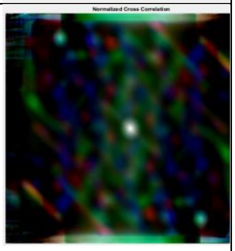
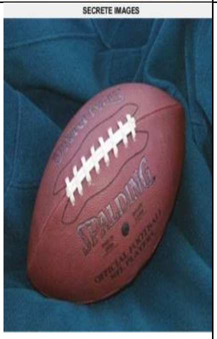
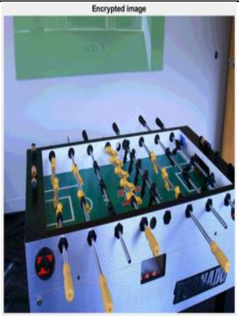
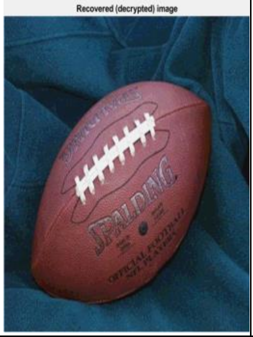
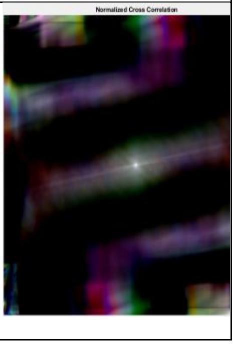
Decoding (Extraction): Decoding involves extracting the secret image from the encoded cover image. This process requires knowledge of the encoding algorithm used and may involve reversing the steps taken during encoding. By analyzing the pixel values or other characteristics of the cover image, the hidden data can be retrieved and reconstructed to reveal the original secret image.



● UNET
○ VNET
○ UNET++
○ Hybrid of UNET,VNET,UNET++

Figure 7 proposed GUI

Table 1 Test samples of U-Net encoder model.

	Foreground	Secrete Image	Encrypted image	decrypted image	NCC
1					
2					
3					

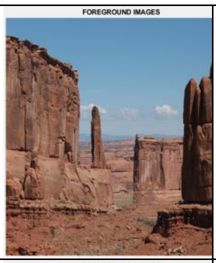
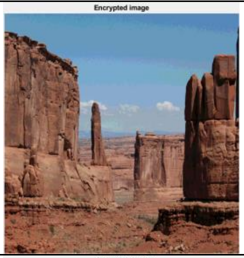
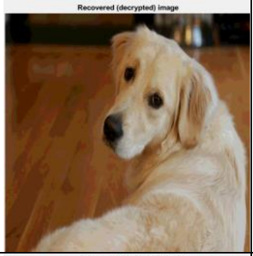
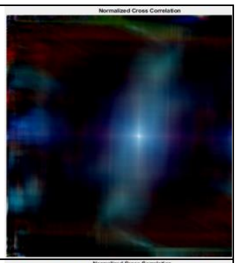
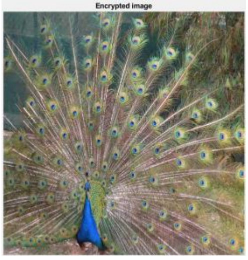
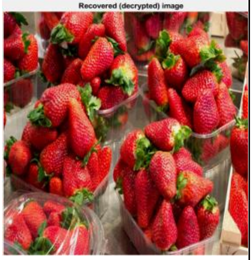
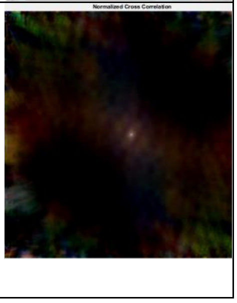
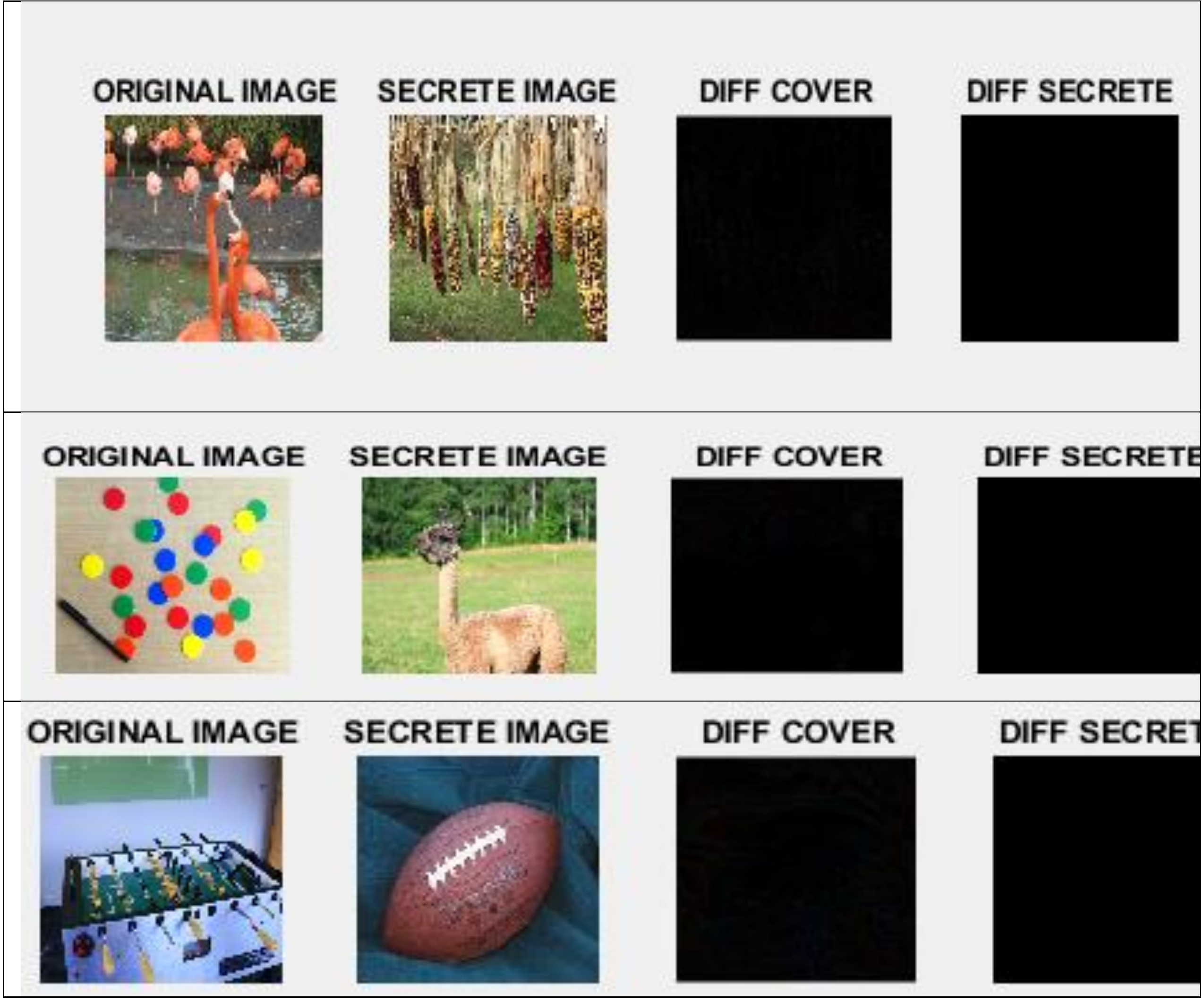
4					
5					

Table 2 performance of U-Net encoder model.



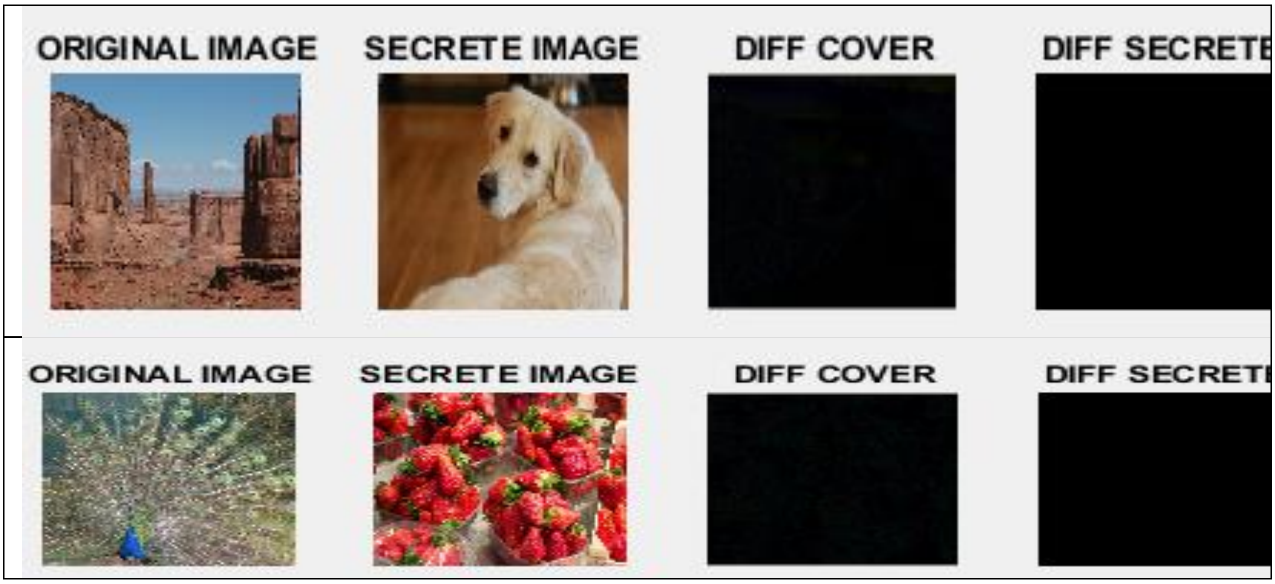
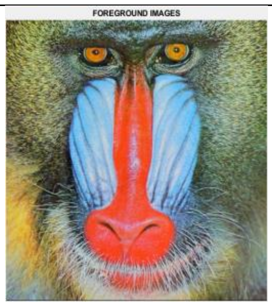
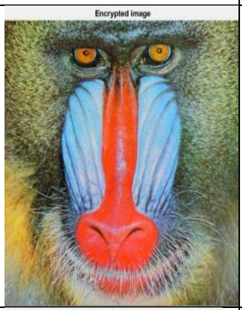
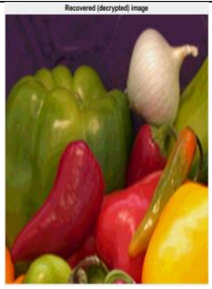
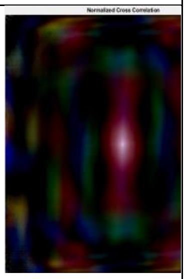
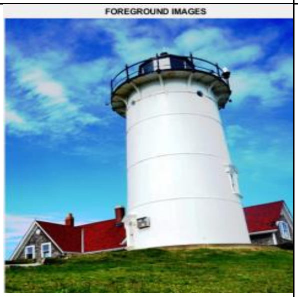
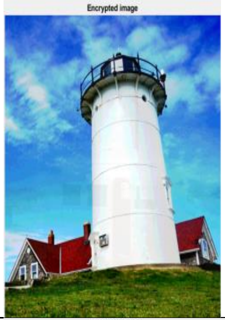
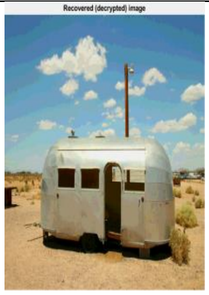
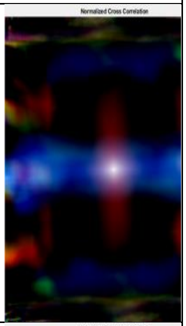
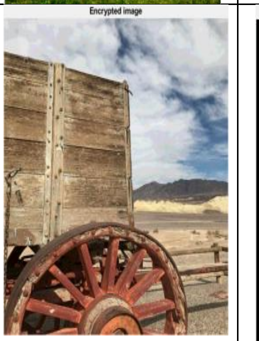
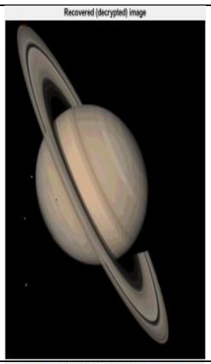
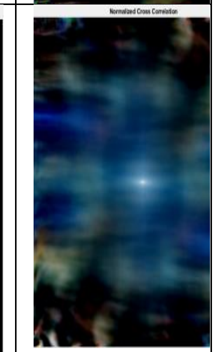
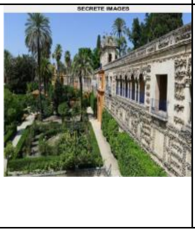
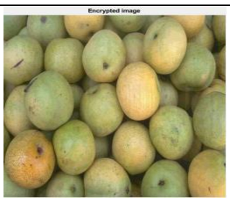
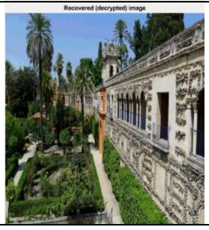
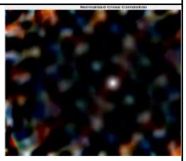


Table 3 Test samples of V-Net encoder model

	Foreground	Secrete Image	Encrypted image	Decrypted image	NCC
1					
2					
3					
4					

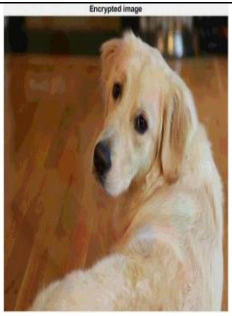
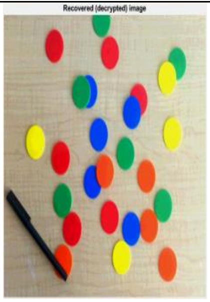
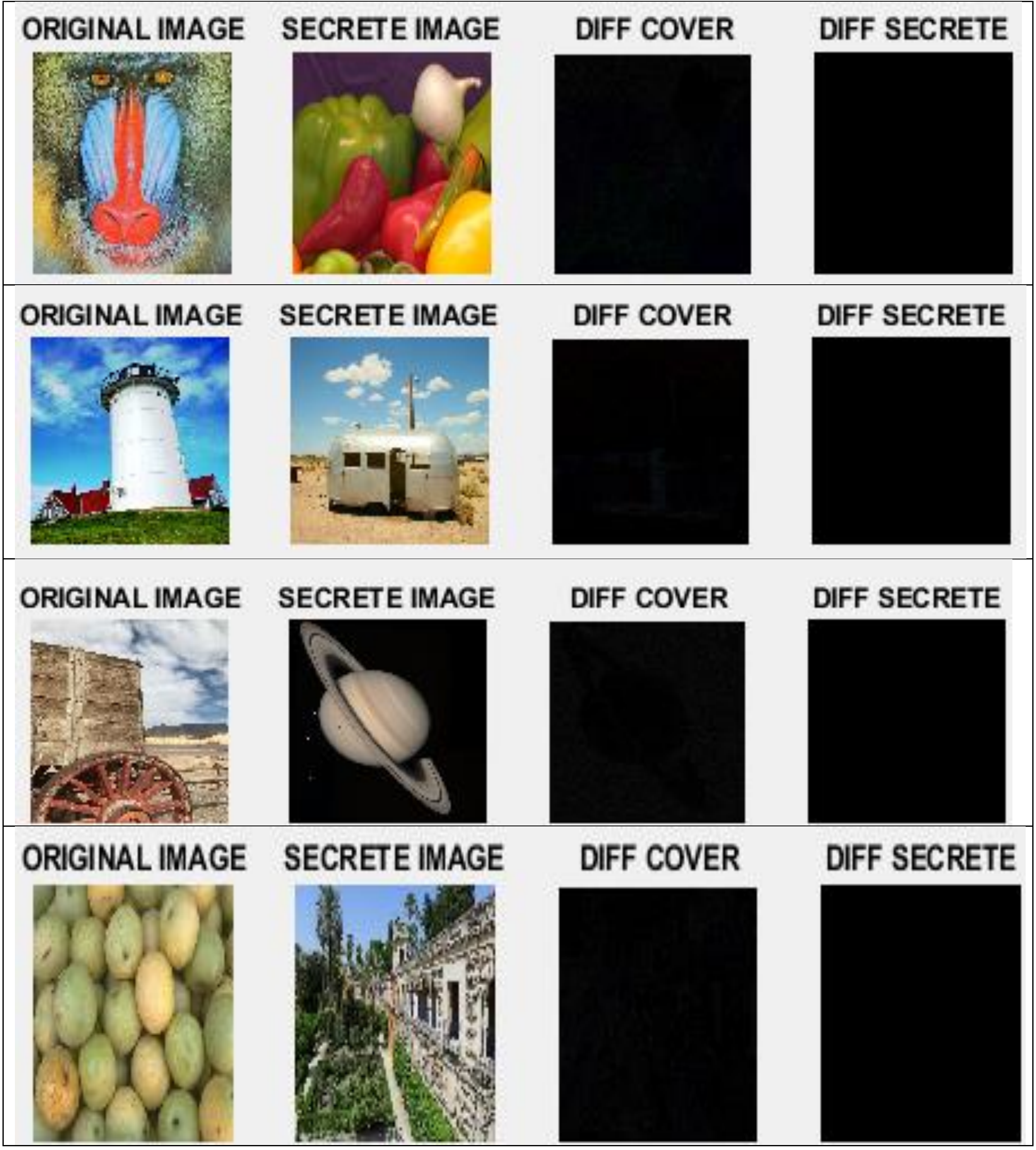
5					Normalized Cross Correlation

Table 4 performance of V-Net encoder model.



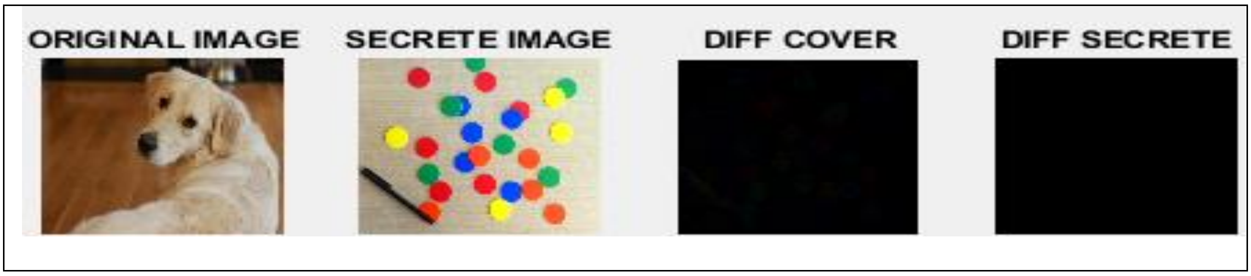


Table 5. Test samples of U-Net++ encoder model.

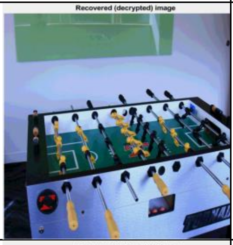
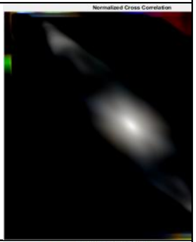
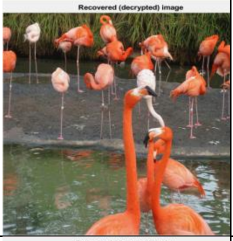
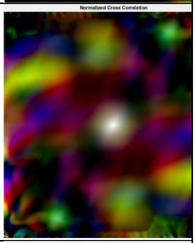
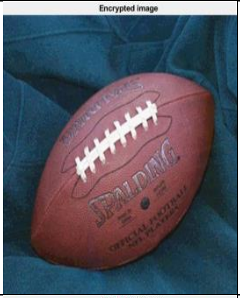
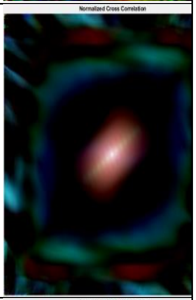
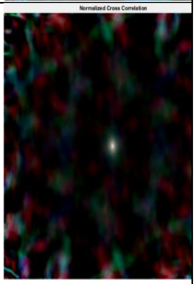
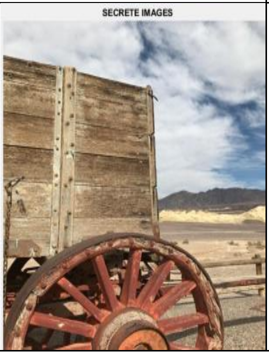
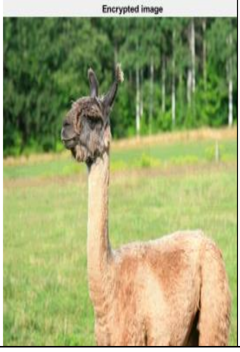
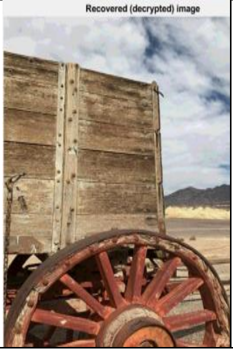
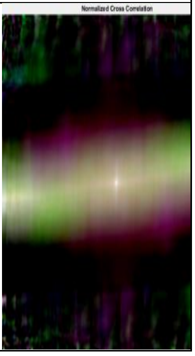
	Foreground	Secrete Image	Encrypted image	Decrypted image	NCC
1					
2					
3					
4					
5					

Table 6. Performance of U-Net++ encoder model.

VI PERFORMACEN EVALUTION

The proposed approach aim to provide a balance between embedding capacity, image quality, security, and robustness to attacks, while being efficient and fast in both the embedding and extraction processes. There are various evaluation metrics that will be used to measure the performance of steganography techniques based on U-Net, V-Net, and U-Net++. the proposed approach aim to improve the quality of the steganographer image by minimizing the distortion caused by embedding the secret information and aim to increase the amount of secret information that can be embedded within an image while maintaining a good level of quality image as well as aim to be efficient and fast in both the embedding and extraction processes, Here are some commonly will be used metrics

Peak Signal to Noise Ratio (PSNR) is a commonly used metric to measure the quality of an image in the context of image steganography. The formula to calculate PSNR is:

$$\mathbf{PSNR = 10 * log10((M^2) / MSE) \quad Eq.1}$$

Where M is the maximum pixel value of the image (usually 255 for 8-bit images), and MSE is the mean squared error between the original and steganographer images.

Mean Absolute Error (MAE): MAE measures the average absolute difference between the pixels in the original and steganographic images. A lower MAE value indicates better image quality. In the context of image steganography, the Mean Absolute Error (MAE) is a metric that measures the average absolute difference between the pixel values of the original image and the steganographic image. The formula for calculating the MAE is:

$$\mathbf{MAE = (1/N) * \sum i=1 \text{ to } N \mid I(i) - S(i) \mid Eq.\ 5.3}$$

Where N is the total number of pixels in the images, I(i) and S(i) are the pixel values of the original and steganographic images, respectively, at the ith pixel location. [40]

Mean Square Error (MSE): MSE is a common metric for evaluating image quality. It measures the average of the squared differences between the pixels in the original and the steganographic images. A lower MSE value indicates better image quality. In image steganography, the Mean Square Error (MSE) is a commonly used evaluation metric to measure the quality of the steganographic image compared to the original image. It is calculated using the following formula:

$$\mathbf{MSE = (1/N) * \sum(i=1 \text{ to } N) [I(i) - S(i)]^2 \quad Eq.\ 5.4}$$

Where:

- N is the total number of pixels in the image
- I(i) is the intensity value of the ith pixel in the original image
- S(i) is the intensity value of the ith pixel in the steganographic image

Visual Information Fidelity (VIF): VIF is a metric that measures the similarity between the original and steganographic images in terms of the amount of visual information preserved. A higher VIF value indicates better image quality. The formula for VIF can be expressed as:

$$\mathbf{VIF=(2*sigma\_xy+C1)*(2*sigma\_xy+C2)/(sigma\_x\_sq+sigma\_y\_sq+C1)/(sigma\_x\_sq+ sigma\_y\_sq + C2) \quad Eq.\ 5.5}$$

Where:

- sigma_xy is the covariance between the original and steganographic images.
- sigma_x_sq and sigma_y_sq are the variances of the original and steganographic images, respectively.

C1 and C2 are constants that stabilize the division in the formula and prevent the denominator from being too small. VIF values range from 0 to 1, with a higher value indicating better visual information fidelity between the original and steganographic images. A VIF value of 1 indicates perfect visual information fidelity between the two images.

Normalized correlation coefficient (NCC) : The normalized correlation coefficient (NCC) is a metric commonly used to measure the correlation between the original and steganographic images in image steganography. The formula for NCC can be expressed as:

$$\mathbf{NCC = (1/n) * \sum(x - \mu\_x) *(y - \mu\_y) / \sigma\_x*\sigma\_y \quad Eq.\ 5.6}$$

Where:

n is the total number of pixels in the images

x and y are the original and steganographic images, respectively

- μ_x and μ_y are the mean pixel values of the original and steganographic images, respectively σ_x and σ_y are the standard deviations of the pixel values of the original and steganographic images, respectively.

Table 7 PSNR performance metrics for Unet,Vnet,Unet++ models

Test sample	UNet,	VNet	UNet++
1.	32.78	31.76	31.50
2.	32.58	32.53	32.78
3.	31.95	30.13	33.69

4.	32.25	32.12	32.82
5.	31.53	31.95	32.40

The table 7 presents the Peak Signal-to-Noise Ratio (PSNR) performance metrics for the UNet, VNet, and UNet++ models across five test samples. The PSNR values indicate the quality of the stego images generated by each model, with higher values representing better fidelity to the original images. For UNet, the PSNR values range from 31.53 to 32.78, for VNet they range from 30.13 to 32.12, and for UNet++ they range from 31.50 to 33.69. These metrics provide insights into the comparative performance of the three models in terms of image quality preservation during the steganography process.

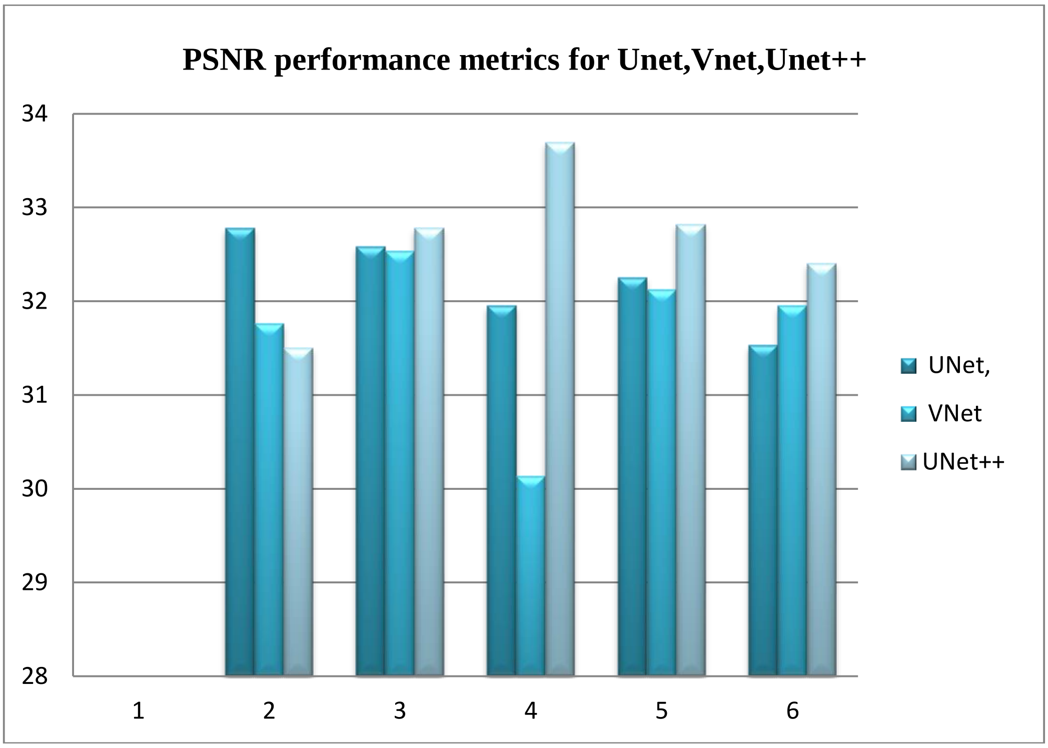


Figure 8 PSNR performance metrics for Unet, Vnet,Unet++ models

Table 8 Mean Absolute Error (MAE) performance metrics for UNet,VNet, UNet++ models

	UNet,	VNet	UNet++
1	0.93	1.09	1.52
2	1.30	1.25	1.42
3	1.60	0.88	1.17
4	1.30	1.47	0.97
5	0.93	1.23	0.64

The table 8 shows the Mean Absolute Error (MAE) performance metrics for the UNet, VNet, and UNet++ models across five test samples. The MAE values represent the average absolute difference between the original and stego images, with lower values indicating better performance in terms of image fidelity. For UNet, the MAE values range from 0.93 to 1.60, for VNet they range from 0.88 to 1.47, and for UNet++ they range from 0.64 to 1.52. These metrics provide insights into the comparative performance of the three models in terms of accurately concealing the secret image within the cover image.

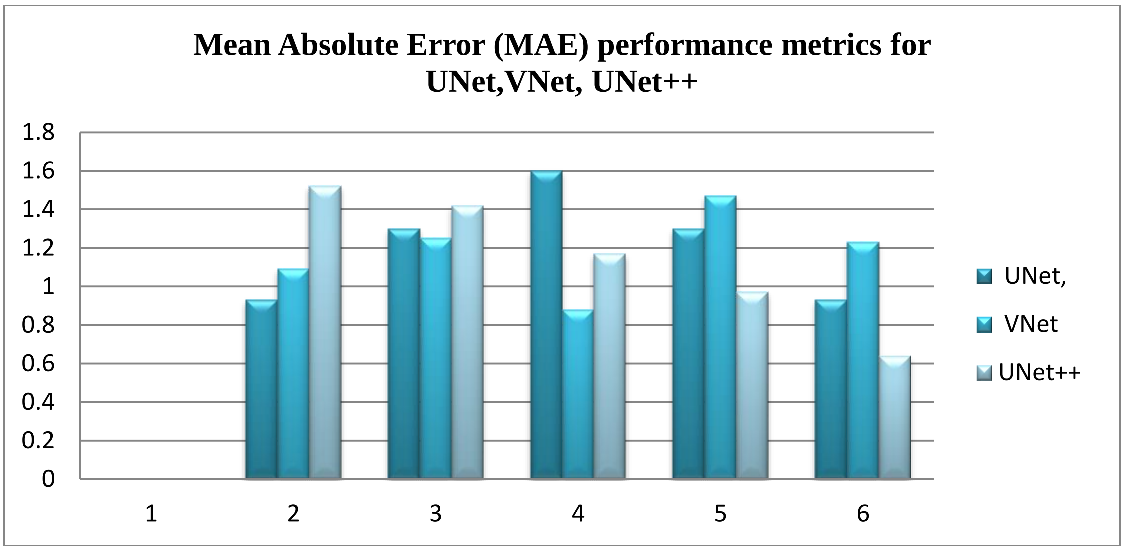


Figure 9 Mean Absolute Error (MAE) performance metrics for UNet,VNet, UNet+

Table 9 Visual Information Fidelity (VIF) performance metrics for Unet,Vnet,Unet++ models

Test sample	UNet,	VNet	UNet++
1	0.09	0.13	0.03
2	0.07	0.06	0.10
3	0.15	0.20	0.09
4	0.11	0.11	0.09
5	0.12	0.04	0.07

The table 9 displays the Visual Information Fidelity (VIF) performance metrics for the UNet, VNet, and UNet++ models across five test samples. The VIF values indicate the preservation of visual information in the stego images, with higher values reflecting better fidelity to the original images. For UNet, the VIF values range from 0.07 to 0.15, for VNet they range from 0.04 to 0.20, and for UNet++ they range from 0.03 to 0.10. These metrics provide insights into the comparative performance of the three models in terms of preserving visual quality during the steganography process.

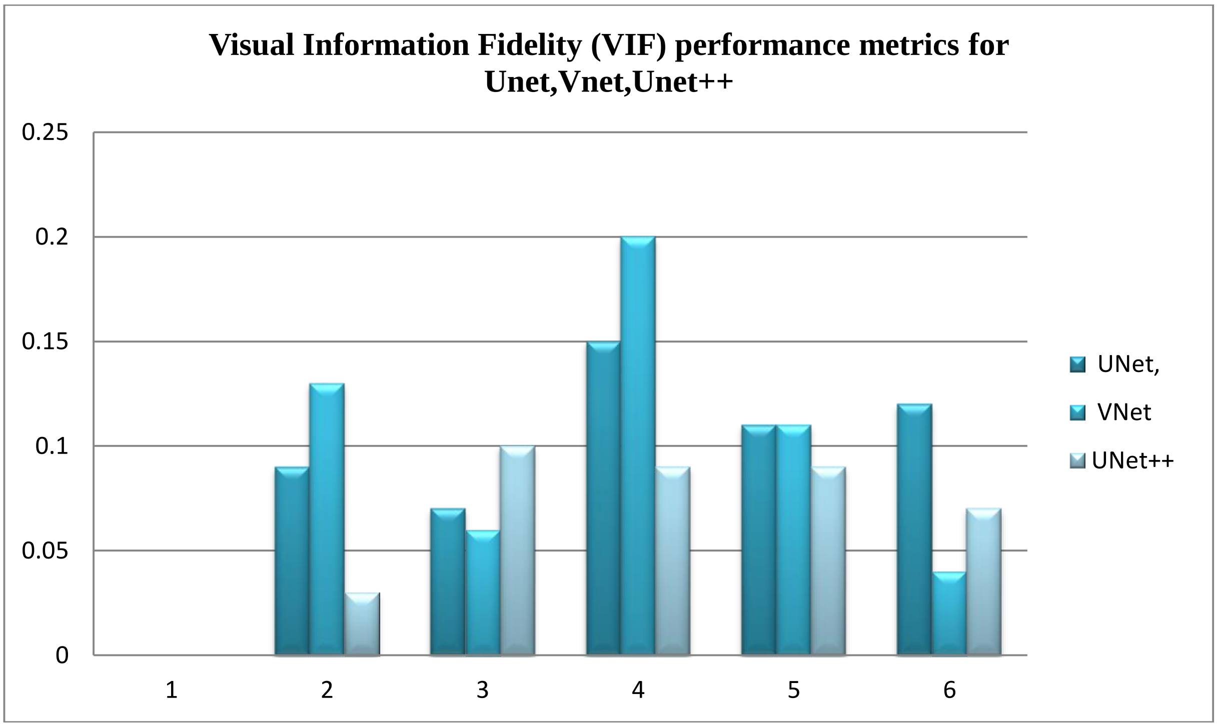


Figure 11 Visual Information Fidelity (VIF) performance metrics for Unet,Vnet,Unet++

VII CONCLUSION

This research study explores the effectiveness of deep learning-based encoders, specifically U-Net, V-Net, and U-Net++, for image-in-image steganography. Traditionally, methods in digital image steganography have employed spatial and transform techniques to embed information within images. However, with the growing concerns surrounding data security, deep learning approaches have emerged as promising solutions to overcome limitations such as low embedding capacity and poor reconstruction quality. The comparative analysis conducted in this research reveals that while all three architectures can effectively conceal a secret image within a cover image, the U-Net architecture outperforms V-Net and U-Net++ in terms of embedding capacity and the quality of stego and reconstructed secret images. This finding provides valuable insights into selecting suitable deep learning-based encoders for securing digital images against unauthorized access during transmission and storage. Overall, the study contributes to advancing the field of image steganography by evaluating and comparing the performance of different deep learning architectures, thereby facilitating informed decision-making in the development of robust steganographic systems.

REFERENCES

1. F.Y. Shih, Digital Watermarking and Steganography: Fundamentals and Techniques, CRC Press, 2017, http://dx.doi.org/10.1201/9781315121109.
2. N. Provos, P. Honeyman, Hide and seek: An introduction to steganography, IEEE Secur. Priv. Mag. 1 (2003) 32–44, http://dx.doi.org/10.1109/MSECP. 2003.1203220.
3. R. Amirtharajan, J.B. Balaguru Rayappan, An intelligent chaotic embedding approach to enhance stego-image quality, Inform. Sci. 193 (2012) 115–124, http://dx.doi.org/10.1016/j.ins.2012.01.010.
4. D. Artz, Digital steganography: hiding data within data, IEEE Internet Comput. 5 (2001) 75–80, http://dx.doi.org/10.1109/4236.935180.
5. A. Cheddad, J. Condell, K. Curran, P. Mc Kevitt, Digital image steganography: Survey and analysis of current methods, Signal Process. 90 (2010) 727–752, http://dx.doi.org/10.1016/j.sigpro.2009.08.010.
6. G. Fornarelli, A. Giaquinto, An unsupervised multi-swarm clustering technique for image segmentation, Swarm Evol. Comput. 11 (2013) 31–45, H.J. Highland, Data encryption: A non-mathematical approach, Comput. Secur. 16 (1997) 369–386, http://dx.doi.org/10.1016/S0167-4048(97)82243- 2.
7. M.S. Subhedar, V.H. Mankar, Current status and key issues in image steganography: A survey, Comput. Sci. Rev. 13–14 (2014) 95–113, http:
8. Min Wu, Bede Liu, Data hiding in image and video. I. Fundamental issues and solutions, IEEE Trans. Image Process. 12 (2003) 685–695,
9. S.N. Mali, P.M. Patil, R.M. Jalnekar, Robust and secured image-adaptive data hiding, Digit. Signal Process. 22 (2012) 314–323, http://dx.doi.org/10.1016/ j.dsp.2011.09.003. [11] F.a.P. Petitcolas, R.J. Anderson, M.G. Kuhn, Information hiding-a survey, Proc. IEEE 87 (1999) 1062–1078,
10. N.F. Johnson, S. Jajodia, Exploring steganography: Seeing the unseen, Computer (Long. Beach. Calif). 31 (1998) 26–34, http://dx.doi.org/10.1109/ MC.1998.4655281.
11. N. Johnson, S. Katzenbeisser, A survey of steganographic techniques, Inf. Hiding (2000) 43–78, http://67.192.244.68/uploads/public/documents/ chapters/Petitcolas035-ch03.pdf.
12. N.F. Johnson, S. Jajodia, Steganalysis: the investigation of hidden information, in: 1998 IEEE Inf. Technol. Conf. Inf. Environ. Futur. (Cat. No.98EX228), IEEE, 1998, pp. 113–116, http://dx.doi.org/10.1109/IT.1998.713394.
13. C.-S. Hsu, S.-F. Tu, Finding optimal LSB substitution using ant colony optimization algorithm, in: 2010 Second Int. Conf. Commun. Softw. Networks, IEEE, 2010, pp. 293–297, http://dx.doi.org/10.1109/ICCSN.2010.61.
14. A.D. Ker, Steganalysis of LSB matching in grayscale images, IEEE Signal Process. Lett. 12 (2005) 441–444,
15. R.-Z. Wang, C.-F. Lin, J.-C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, Pattern Recognit. 34 (2001) 671–683, http://dx.doi. org/10.1016/S0031-3203(00)00015-7.
16. D.-C. Wu, W.-H. Tsai, A steganographic method for images by pixel-value differencing, Pattern Recognit. Lett. 24 (2003) 1613–1626, http://dx.doi.org/ 10.1016/S0167-8655(02)00402-6.
17. R.J. Anderson, F.A.P. Petitcolas, On the limits of steganography, IEEE J. Sel. Areas Commun. 16 (1998) 474–481,
18. S. Lee, C.D. Yoo, T. Kalker, Reversible image watermarking based on integer-to-integer wavelet transform, IEEE Trans. Inf. Forensics Secur. 2 (2007) 321–330,

19. Baluja, S. Hiding images in plain sight: Deep steganography. Adv. Neural Inf. Process. Syst. 2017, 30, 2069–2079.
20. Baluja, S. Hiding images within images. IEEE Trans. Pattern Anal. Mach. Intell. 2019, 42, 1685–1697.
21. Zhu, J.; Kaplan, R.; Johnson, J.; Fei-Fei, L. Hidden: Hiding data with deep networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 657–672.
22. Luo, X.; Zhan, R.; Chang, H.; Yang, F.; Milanfar, P. Distortion agnostic deep watermarking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 13548–13557.
23. Qin, J.; Wang, J.; Tan, Y.; Huang, H.; Xiang, X.; He, Z. Coverless image steganography based on generative adversarial network.Mathematics 2020, 8, 1394.
24. Shang, Y.; Jiang, S.; Ye, D.; Huang, J. Enhancing the security of deep learning steganography via adversarial examples. Mathematics2020, 8, 1446.
25. Zhu, X.; Lai, Z.; Zhou, N.; Wu, J. Steganography with High Reconstruction Robustness: Hiding of Encrypted Secret Images. Mathematics 2022, 10, 2934.
26. Wang, Z.; Feng, G.; Wu, H.; Zhang, X. Data hiding during image processing using capsule networks. Neurocomputing 2023,537, 49–60
27. Zhang, C.; Benz, P.; Karjauv, A.; Sun, G.; Kweon, I.S. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. Adv. Neural Inf. Process. Syst. 2020, 33, 10223–10234.
28. Chen, F.; Xing, Q.; Fan, C. Multilevel Strong Auxiliary Network for Enhancing Feature Representation to Protect Secret Images.IEEE Trans. Ind. Inform. 2021, 18, 4577–4586. [CrossRef]
29. Wang, Z.; Zhou, M.; Liu, B.; Li, T. Deep Image Steganography Using Transformer and Recursive Permutation. Entropy 2022,24, 878.
30. Akshay Kumar[a,*], Rajneesh Rani[a], Samayveer Singh(2023) Encoder-Decoder Architecture for Image Steganography using Skip Connections. International Conference on Machine Learning and Data Engineering 10.1016/j.procs.2023.01.091
31. Tancik, M.; Mildenhall, B.; Ng, R. Stegastamp: Invisible hyperlinks in physical photographs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 2117–2126.
32. Wengrowski, E.; Dana, K. Light field messaging with deep photographic steganography. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1515–1524.
33. Hayes, J.; Danezis, G. Generating steganographic images via adversarial training. Adv. Neural Inf. Process. Syst. 2017,30, 1954–1963.
34. Zhang, C.; Benz, P.; Karjauv, A.; Kweon, I.S. Universal adversarial perturbations through the lens of deep steganography: Towards a fourier perspective. In Proceedings of AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; pp. 3296–3304.
35. Jung, D.; Bae, H.; Choi, H.S.; Yoon, S. Pixelsteganalysis: Pixel-wise hidden information removal with low visual degradation.IEEE Trans. Dependable Secur. Comput. 2023, 20, 331–342.
36. Xiang, T.; Liu, H.; Guo, S.; Zhang, T. PEEL: A Provable Removal Attack on Deep Hiding. arXiv 2021, arXiv:2106.02779.
37. Zhong, S.; Weng, W.; Chen, K.; Lai, J. Deep-learning steganalysis for removing document images on the basis of geometric median pruning. Symmetry 2020, 12, 1426.
38. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks.arXiv 2013, arXiv:1312.6199.
39. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. arXiv 2014, arXiv:1412.6572.
40. Chen, H.; Zhu, T.; Zhao, Y.; Liu, B.; Yu, X.; Zhou, W. Low-frequency Image Deep Steganography: Manipulate the Frequency Distribution to Hide Secrets with Tenacious Robustness. arXiv 2023, arXiv:2303.13713.
41. Sapna Kaneria, Dr. Varsha Jotwani (2024), Advancements in Digital Steganography: A State-of-the-Art Review, IOSR Journal of Computer Engineering (IOSR-JCE); e-ISSN: 2278-0661, Volume 26, Issue 1, Ser. 1 (Jan. – Feb. 2024), pp 49-62. doi: : 10.9790/0661-2601014962.