International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 12s,2025
https://theaspd.com/index.php

Video Steganography Model Based on a Deep Neural Network

Swagata Sarkar[1], Janani Selvam[2], Divya Midhun Chakkaravarthy[3]
[1]Post Doctoral Fellow,Lincoln University College,Malaysia,sacademiczone@gmail.com
[2]Faculty of Engineering Lincoln University College,Malaysia,vijayjanani.s@gmail.com
[3]Faculty of Engineering Lincoln University College, Malaysia,divya@lincoln.edu.my

Abstract
This article discusses about Stego-STFAN, a novel way to hide the contents inside a video. A model method that doesn't cost much connects the time and space areas. The Stego-STFAN method gets 27.03 and 23.09, which are close to the best that can be done now. It is called steganography to hide a message so that only certain people can read it. When you use steganography, one thing to consider is how big the message you want to hide is. This is because as the message gets longer, the protection level drops. They got bigger because of this thought. A new way to hide movies is based on the Attention process and the Spatial-Temporal Adaptive Filter Network (STFAN). This will help stego-frames work better and get more secrets back. Handling frames this way's faster and better because the filters and maps change as the frames are played.
Keywords: *Traditional Image Steganography, Steganography, General Architecture, Attention, Group Normalization Block.*

1. INTRODUCTION

You can hide messages in your cover photos with picture steganography. With a steganography picture, you can send a secret message over a public road. The secret word might be easy to figure out for someone who gets the message. There are many ways in which the new ways are better than the old ones. This is especially true for CNNs, which stand for convolutional neural networks. That's a shock. In the past few years, this happened. There were early versions of Alex Net, VGGNet, and later, more advanced versions of Inception (Krizhevsky, A. et al 2012). Everything was going on at the same time. People have also used deep learning to hide things. Different kinds of steganography aren't as good as deep learning steganography. You can now hide notes in the top Picture so others can't see them. Deep learning allows you to send longer letters that hide things. You can still hide texts with pictures using lossy deep steganography. CNN was shown. A coder and a decoder were built into it. A cover picture the same size as the secret Picture can hide one secret Picture. After that, the decoder network can show the secret Picture (Simonyan, K. et al 2015).Deep learning can also be used to make a tool for steganography that looks like CNN. This way is faster and better than the one we have now. We made it with deep learning by adding generative adversarial networks (GANs) to CNN networks. No one else could read the secret letter, so it was safer. We tried to make it as close to seeing a nature picture instead of a stego picture. In studies of steganography that have already been done, pictures of papers have not been used to hide a message. Pictures of papers can be sent as text messages, but not Pictures of people. So, social networks should keep users' information safe and check that papers are correct (Szegedy, C. et al 2015). A message is hidden in this study by a picture of a piece of paper. You can use the Picture at school or in your daily life. The structure similarity index measure (SSIM) and the peak signal-to- noise ratio (PSNR) are the two main ways to rate how good a picture is. In fair and full ways, these can rate pictures, but they don't tell you how well the pictures were made. One part of this study project checks the document image quality (DIQA) to see how well the secret message was removed from the document's Picture.Steganography is the area of study that looks into ways to get around it. It is shared, and people who listen are sent pictures with hidden information written on them and told to look at them. Steganography and steganalysis have helped each other since the start of the study. Steganography lets you find or get rid of messages that were hidden (He, K. et al 2016). This method is called step analysis. Also, it's hard to tell if there is secret information there, and it's even harder to get that information out if you don't know how steganography works. It is now essential to study how to get rid of steganography. So far, deep learning steganography

has only been used a few times to see how well it can hide information. It's called PixelSteganalysis. That's what it was based on. People get mad because there is a message hidden there. Cases in the news show how to see this (Xu, Shutong, et al 2022). Getting rid of bad results is the only way to get rid of deep-learning hoaxes. They helped us develop a deep-learning way to eliminate steganography for this work.You could call it ComDefend because it would be like a picture that goes from beginning to end. One way to avoid sample scams is to do this. It has two different types of neural networks. It makes a good copy of the original Picture (Baluja, S. et al 2017), and ComCNN remembers how it is assembled. The study's results led to the idea of using deep learning to get rid of steganography. There are strong disturbance reduction methods and deep-learning steganography used in this model. It's bigger and can be used in lots of different ways. When you say "embedding capacity," you mean how much data can be put into a digital movie without making it less valuable. There are, however, many more ways to hide steganography now than there were in the past. This cover can hold PDFs, pictures, and movies you can watch online. Videos are better for hiding things than pictures because they show more of what is there. Remember that studying a deep learning-based video steganography method is still very new. There are not many studies that can be used right now.It worked out much better to hide almost nothing but the source frames. Wang learnt new things and used them to make a CNN that keeps an inter-frame record hidden in a cover movie (Goodfellow, I.J. et al 2014). The layer hiding movies now has an extra layer that can be changed differently to improve the Network. This kind of movie is also called a "stego video," you can use DVMark to make it. It works with H.264 quality files. It was based on GAN and was yet another way to hide the whole movie. More attention was paid to it so encoders and decoders could get better at pulling features. Watching stego videos is better because the picture quality is better, and the extraction rate is faster. But they can only compress videos in one way, and it's not the most common way. The PyraGAN that Chai showed is given the CU faces. Besides that, this tells the Network more about how hard the movie's pictures are (Subramanian, N. et al 2021). This method, however, did not check how stable the system was for small video files. You could also hide the whole movie, even when H.264 compression was used. It used both timing and motion capture to ensure the movie was well-made.How to avoid compression is also very important if you're interested in end-to-end steganography.How things are done now could help get us out of this mess. HiDen made a differentiable compression layer that shrunk the Picture with the help of the discrete cosine transform (DCT). It was a lot like how compression was used before. The quantisation steps were changed with one-step functions so that they were close to zero. This was done to get as close as possible to the DCT numbers that can't be changed in JPEG quantisation. This helped us understand what a round shape is (Xu, S. et al 2022). A different deep-learning method with two steps could be used to add watermarks to pictures. In the first step, the encoder and the decoder learn how to work without JPEG compression. Here is the next step. A new JPEG file will make the stamp picture smaller before sending it to the printer. This will help them see the mark on a squished picture. There are two steps to this method. You can teach the Network immediately instead of going through the step where it can't tell the difference. But, the encoder is the only one who can learn how to compress (Weng, X. et al 2018).And its name was Mini-Batch of Real and Simulated JPEGs. It was used to show the noise layer. There were three steps: real JPEG, virtual JPEG, and real JPEG (to show no compression). This noise layer had to fix the problems already brought up. Each batch has a different order of these three states.People who send and receive info can learn how to narrow it down this way. There was a container enhancement module (CEM) that could be used to improve the stego Picture instead of trying to blow it up. This module does the same thing as the JPEG module at the beginning of the encoder. Being able to bend less isn't so bad when this thing is there. Add a new layer that works this way to shrink a video file. Five kinds of noise were present: scale noise, Gaussian noise, middle-level noise, and light and heavy JPEG compression. You can change how much each type of noise weighs. The trick is to mix the noise to make it look more like videos that have been sped up. It was already said that a compression net, which is a small 3D CNN, works like video compression. A 3D light model was used to make the distortion network.This was done so the video wouldn't twist when it was taken and then shrunk. Setting up an entire video steganography network is challenging (Luo, X. et al 2021). For

steganography to work, the text must be straightforward, and both the input and output rates must be high. Because of this, it is very important to have a deep learning network that can keep and read data and work well. Second, learn how to use the function to see how much time has passed in different movie scenes. This is because pictures and photos use time in various ways. Because most films only use one type of code, people who use steganography must also consider how to keep movies from being shrunk. How does the drop work? It's not unique, so it can't go straight to the deep learning network. How to teach the Network to shrink is one of the most important things to think about right now. We can now show you a wholehow to hide movies that use a multi-scale network and different types of video compression. In our app, video is the first way to send a text message. This was an easy choice for us because movies are well-known and show a lot of information over and over.It's great that video can also hide things. The main plan for the Network, GAN, shows how all of its parts work. This basic network system comprises an encoder, a decoder, and a discriminator (Wiegand, T. et al 2003). The discriminator and the encoder-decoder are competing with each other. It's good that Stech doesn't work as well with this tool. There are now pyramid-shaped nodes in both the encoder and the decoder. It might be easier to find traits if the Network is set up like a tower and data is sent at different speeds. Also, most of the Network's convolutional layers use three-dimensional convolution to get and use the movie's time data properly. It's hard to keep the movie from getting squished. We start by adding a noise layer to the Network. This is done when there is noise between the encoder and the decoder (Shrestha, A. et al 2015). We call the changes that happen when you squeeze film "noise." Many people use end-to-end video steganography, which means that tests show it to work well. You can also shoot well. That's what we saw on the network plan before. Finally, it can handle several well-known ways to reduce the size of a video file.It has always been essential to send things without being seen. Over the years, secret contact has grown. This field of study investigates hiding private information in a picture to look like it was never changed. It's been getting a lot of attention lately. You can set safety rules and save information here. Now that the world is connected talking to each other is safer and more functional. Pictures can hide many things, like private and business problems, to keep people and the country safe (Weng, Xinyu, et al 2019). Bad people steal other people's work and give it to other people to use. Picture steganography has been used in more places over time because of this double-edged sword. We need to find pictures that hide things in this area. It's not easy to find picture steganography. Finding steganography that is hard to figure out takes more work. When stereoscopy works well, there aren't many signs that the messages have changed. There aren't as many hidden signs to find. People look at data and use plots all the time. People who study numbers look for strange patterns in how pixel numbers are spread out (Holub, V. et al 2012). This is a different kind of attack that hurts people by making them look for mistakes in pictures.A lot of people act like this. A lot of work went into making these ways, but they don't always work or help with codes that are hard to read. There is more picture steganography now that AI is more well-known. There may be more ways to find secret material now that AI is used more. This is because AI can learn to notice the little changes steganography makes. A lot has changed with "deep learning" AI. CNNs can tell the difference between little changes in the way things look. A lot of different AIs can catch people who try to hide things. This isn't the best way for AI to find hidden pictures. There aren't enough big samples that have already been marked to train with. This is even more true since these records are kept in a secret location that is hard to reach (Li, B. et al 2014). These methods are also bad because they make computers work harder. These "black box" AI models make it hard to see how they decide what to do. People are curious about how strong and open the system is. The best things about AI are its strength, flexibility, and breadth. We will also discuss how it could be used to find picture steganography.We talk about how far this field has come. Follow these steps to fix old forms of ID that are giving you trouble. We also talk about its problems and try to guess what changes might be made in the future to help us deal with these problems better. It's helpful to know how to spot and understand picture steganography when privacy and data security are paramount (Holub, V. et al 2014). AI can help us get there faster. We could stay in touch with the web. In the long run, it's better. SSIS, which stands for spread spectrum image steganography, is a terrific way to hide pictures. SPC is used in many radio broadcasts because it can make lines that are safe, strong, and

free of noise. That's what this method is based on. The picture that makes it look like the information is spread out in many places hides secret information from the SSIS. They look like noise now. With this method, the secret data is spread out over an extensive frequency range. CCTV cameras can't distinguish between noise and saved data, so it's hard to find. Some think these AI-based ways to find things are the most cutting-edge (Marvel, L.M. et al 1998). This is what our CNNs are made of: "Convolutional Neural Networks." Any AI that finds pictures is also good at this tricky part of AI. Our research shows that AI-based recognition methods, especially CNNs, are usually better at finding secret data. One way to do this is to take a lot of tests. We still had a hard time finding SSIS. Even the best search engines can't find it this way.

2. RELATED WORK

Changes are made to the cover image's least significant bits (LSBs). This is what most types of picture steganography do. Something important might be hidden here. AI has been used to make systems that hide pictures in the last few years. These hiding pictures might show more things and look better than other methods. On the other hand, many deep learning systems see grounding and extraction as two different tasks. There's a chance that this means the stego Picture and the secret details are always less clear (Westfeld, A. et al 2001). Several studies look at picture steganography as a problem that can't be solved and use Invertible Neural Networks (INNs) to get around this problem. Picture steganography doesn't work well with straight lines, so both the encoding and extraction steps use a structure that is the same on both sides. This is a good way to fix it. Normal methods, deep learning methods, and methods that use IoT networks as their major building block are all talked about in this piece. I hope it helps you.

2.1. Traditional Image Steganography

Digital watermarking and steganography are the two main ways that data is usually hidden in digital pictures. There are two kinds of ways, depending on how much information is hidden and how the cover picture is connected to what's inside. Bits related to the Picture's information are usually used to do this. This is done to protect copyright or show who owns the Picture. On the other hand, steganography hides the stego image behind the cover picture. This way, many people can talk to each other without being seen. Two more groups that can be used to talk about how to hide things are the spatial-domain techniques and the frequency-domain techniques. They can put these groups together with the help of the secret message. When you work in space, you change the numbers of the images right away (Yu, C. et al 2024). To change something, the n most important parts of the secret message are often put in place of the n least important parts of the cover picture. This is known as the least-significant-bit (LSB) method. Some people said that pixel-value differencing could be used to make the package bigger, while Pan suggested that the values of pixels be changed with panels. Both of the ideas were meant to give people more of it. He could hide data by leaving some pixel numbers alone.

2.1.1 Deep Learning-Based Steganography

Study after study has used this way to look at pictures. Steganography comprises two main networks that work together to make it work from beginning to end. It gets the cover picture and secret message and saves them. After that, it reads the stego Picture and finds the secret message that was hidden inside. To ensure the hidden message is found properly, there is a "reconstruction loss" and a "secret loss." The stego Picture looks a lot like the other cover image. From the beginning to the end, it tells these networks how to be taught. They are taught to a group. A discriminator makes the stego Picture look even better (Ruanaidh, J. et al 1996). Because of this, it is hard to tell the difference between this and the main Picture. This is one more type of training that is used. The Generator Adversarial Network (GAN) idea was used to make it. The discriminator, decoder, and encoder are all taught simultaneously in a way that looks like a game. This kind of steganography can hide A lot of data, and it doesn't change how the Picture looks or how well the data can be recovered (Su P-C, et al 2024). They could hide a full-color picture inside a bigger picture the same size. It was a very secret code. Real pictures from ImageNet are used to teach the system what to do to improve the secret Picture. There is a new way to teach these lessons.

2.1.2. Steganography Based on Invertible Neural Networks

When you use x=$f\square(x)$ in the forward process, you get a number called y. When you use x=$f-1\square(x)$ in the backward process, you get x back from y. The forward and backward processes are the names for these two steps. Both the forward process ($f\square$) and the backward process ($f-1\square$) have the same parameters ($\square$). The flow-based generation model got an affine coupling layer and an invertible 1×1 convolution layer as part of the study. This made it better than the suggested coupling layer. In picture steganography models like ISN, INNs are used. These models see fixing and extracting as problems that can be solved either way (Wang, H.J.M. et al 1998). Since INNs can be turned around, these models use that to make it possible to hide pictures in cover shots that are the same quality. This makes sure that the picture quality stays the same while the secret Picture is being taken. There is a big difference between ISN and Hi Net. Hi Net makes the model work better by adding an extra discrete wavelet transform (DWT) to the raw pictures to change them to the frequency domain. This is the biggest difference between them. The idea that Yang et al. came up with for picture steganography has been changed and is now called PRIS. This design's adding and taking away steps have attack blocks built in.

2.2. Steganography

here are three types of methods for standard picture steganography (Barni, M. et al 2001): those that work in the transformation domain, those that work in the spatial domain, and those that can be used in other areas. The words that describe them come from the ways and places that steganography can be used. When working with space, most of the time, the least significant bit method is used. The series of bits is saved this way in the bottom k important bits of the cover picture. This way, on the other hand, makes adding effects to stego photos simple, and the steganalysis program can tell if the pictures are stego photos or not. People think that hiding data in the transformation domain could be a way to fix the issue of weak resistance and low security spatial are often used to move pictures from the domain of space to the domain of change and then back to the domain of space after the message is added (Zhu, J. et al 2018). There is a better way to hide information in the transform domain than in the spatial domain. It is better at keeping information hidden and strong enough to handle being found.With adaptable steganography, secret information is hidden in places where lines and backgrounds stand out more. This makes the stego Picture look better and less likely that someone will find it. To do this, the parts of the cover picture that can be seen are looked at. It's now much better than it was a few years ago (Luo, Z. et al 2023). Many experts are now using deep learning networks to look into photo silence. One idea was to show a network from end to end how to hide pictures. You could also make stego pictures and find information that had been hidden by making an encoder and a decoder. Black-and-white pictures were hiding inside colour pictures of the same size and quality. It wasn't like how steganography is usually used to hide things. This made it easier for them to hide, but the stego pictures they took changed colour clearly (Liu, J. et al 2020). Afterwards, Hayes pushed for the end-to-end Network to get a discriminator that could distinguish between a stego image and a cover picture. He also said that two-way training could help the author and the discriminator do their jobs better. What was learnt from an earlier study was used to make a new process. This method can trick experts with the help of convolutional neural network design and hiding private data. This would make things safer.

2.2.1 Invertible Neural Networks

Over the past few years, experts have been paying more and more attention to invertible neural networks because they work so well. A neural network that can be turned around maps a hard distribution P x to a simple latent distribution P z in this way. It is possible to cancel the changes that the Network makes. Next, the neural Network determines how the P x and P z distributions are linked (Li, F. et al 2022). The forward process gets complicated data (x) with many factors and returns simple data (z) that fits a distribution. To make a generative model, you go the other way. It is filled with chosen data z to make the complicated data x with many edges. This is the first study to discuss neural networks that can be turned around. An affine link layer in all these networks based on the Real NVP model. Convolutional layers were also added to the linked model to meet the needs of jobs that process images (He, K. et al 2016). Glow is a simple type of created stream that was made with invertible neural networks. A 1x1

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 12s,2025
https://theaspd.com/index.php

convolution was used in this stream. It could be turned around to make very large pictures that looked real to the user.The invertible neural Network splits data into two parts when data comes in. Then, an unknown mechanism changes these parts and connects them in a new way every time. That's the main idea behind how these brain-like computers work. It's not even necessary for the random function to be able to change itself. We've already talked about what a deeply invertible network is made of. You can change three things about neural networks. Two beams make up the map from input to exit. The input and output networks must be the same size, and the Jacobian of the Network can't be 0. Invertible neural networks have been used in many studies to find ways to hide pictures (Ahmadi, M. et al 2020). Scientists thought that a way based on learning might make stego pictures last longer while still letting them store a lot of data and remain invisible. A wavelet transform and invertible neural networks were used to hide a picture with only one colour. Two networks were used separately, which is how embedding and extraction are normally done. This time, the Picture looked better and the extraction method worked better.

3. METHODOLOGY

The method was then broken up into five different parts. A rough sketch of the building being shown was made at the start. So, an explanation of STFAN was written to show why this kind of code was included in the plan that had been laid out. After that, more information was added about why attention blocks (NLSA, NLCA, and CBAM) were used. A short explanation of the DWT and its use in the Network comes next (Vergara, G. F. et al 2024). There is then a line that talks about the loss function that was used. After that, the last fears about the training and how it was done were resolved.

3.1. General Architecture

Three main parts make up Stego-STFAN. In this order, these parts hold the frames in place, hide them, and then show them again. The stego frame (Bt), which you can see in Figure 1, is built at this point. First, we change the cover and secret to YUV. After that, we go through all the subbands again to find the most important parts of the code. The last thing to do is to add the frames for High-Low (HL) and Low-High (LH). Finally, apply the discrete wavelet transform (DWT) to Y and return the frame to RGB.

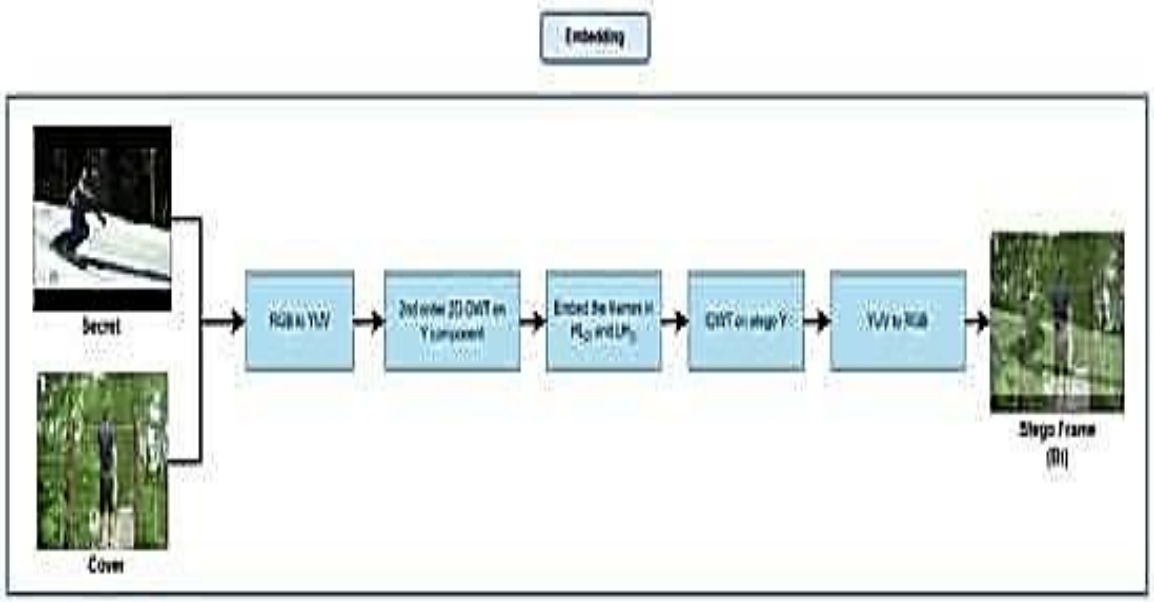


Figure 1. DWT embedding.

The second secret part comes from five places shown in Figure 2. We look at five different types of attention: NLSA, NLCA, the current container frame ($HVt-1$), the most recent frame ($H\mathrm{B}t-1$), and the present stego- frame from the discrete wavelet transform ($H\mathrm{B}t$). Four sources are used in the second step, called the repair part. The box, the expected secret (X−1), the frame before this one (X−1), and this one

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 12s,2025
https://theaspd.com/index.php

(X−1) are these.

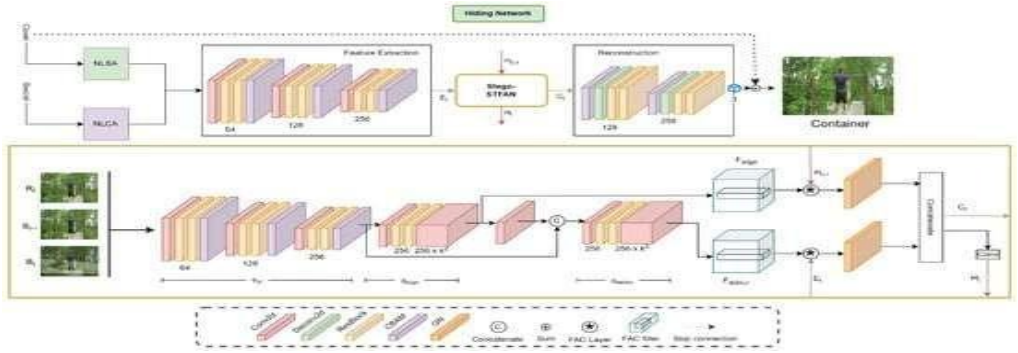


Figure 2. Hiding Network.

These two networks have been added to the block to make this part of the Network work better for extraction. The job of this block is to figure out the best is what makes a map possible in the Non-local way. To learn hard things like lines, pixels that move quickly (these are called "Latent Frame frames"), and the features of the frames' surroundings, this Picture is used. The second part of the secret block is here. Setting up the adjustable filters was the whole point of the first part. Based on these filters, the secret frame changes the little things that make the connections between the time and space parts of the frames stronger. This part has three sources: $\boldsymbol{TVt}$ (-1), $H\mathrm{B}t$ (-1), and $H\mathrm{B}t$. How the Network is put back together is the last part. It's because of how the two parts of the secret block that came before it get mixed up (thanks to FAC layers). The box is moved to the block where the features for extraction are set. There is a secret message that you need to get back. You need to figure it out. This block is in the Network area that heals (Figure 3). You can also find these settings in the second part of the secret block.These filters make it easy to link the time and space parts of the pictures. They also change little things that help them return to the secret message frames more quickly. This part comes from $VVt-1$, $V\mathrm{B}t-1$, and $V\mathrm{B}t$.

How the Network is put back together is the last part. FAC levels come from how the two parts of the previous repair block twist around each other.

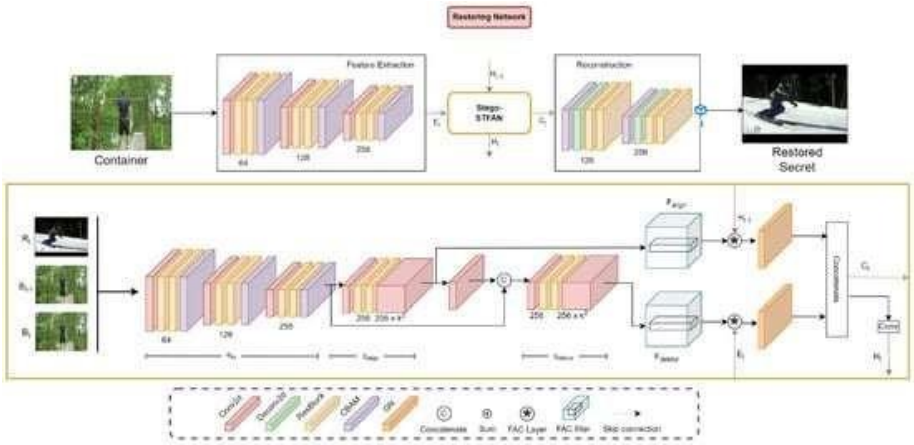


Figure 3. Restoring Network.

3.2. Spatial-Temporal Filter Adaptative Network–STFAN

which is what the organisation suggestion was based on. Their job was to improve the STFAN way of making films. This Network was set up so that it can be used in secret. Each frame can deal with facts about both time and place in real time. The message can be hidden in the top frame and then shown again. Also, STFAN's repeating method makes adding more temporal samples (previous frames) easy and cheap. There's no reason why adding more work to the time part would make things worse. It was already said that the discrete wavelet transform (DWT) gives the secret part the most recent frame (P_1), the previous container frame (P_1), and the current stego-frame (P_1). Thanks to the Filter Adaptive Convolutional (FAC) layer in STFAN, it can mix input from different frames in real time. Other words,

each Picture can be changed on its own. This lets one that is more accurate happen. The Alignment Filter Generator (AFG) and the Concealing Filter Generator (CFG) were the parts that made a difference.The things $yiNc$ and $\boldsymbol{T}Vt-1$ make up $yiNc$. This mix helps you hide the Picture better by showing you where it is. You can also hide things better because the container being built now fits better with the one that came before it. Putting HB$t-1$ and HBt together will also give you $\boldsymbol{bblibn}$. After that, we can get time traits that make it easier to see how the frames fit together. If you know this, you could also make a better container that changes in a good way based on how things move in the movie and has less info that isn't needed. X and Y look at each frame and the bits of early, accurate estimates that deal with space. This is because the prediction changes with each frame. In the same way, the assumed secret (Xi-1) is given to the part that fixes things, along with the old frame (Xi-1) and the new container (Xi-1). STFAN mixes data from different frames in real time to make unique changes to each frame. Now people can make up lies that they are more likely to believe. I know where to hide the truth better because of what has been said.There is a Restoring Filter Generator (RFG) that tells you what the recovered secret should be. This needs to be mixed with the recovered secret. The Network can now easily learn more about space in depth. From what I thought before, it also lets you make a movie that runs better. This filter is made by putting together$V$B$t-1$ and VBt. The part that is healing is then given N$_x\boldsymbol{Tibn}$. You can see how the frames differ when you split out time traits. It also helps you find the secret faster since it's based on the last Picture. The Network is not expensive, and it knows how to handle time and project the right new filter for each frame. It comes from what has been shown and the parts of STFAN that are always the same and different for every sample. You have a better chance of making a change that will let you hide and find the message. It should work better in pictures that move quickly and hide the message more if there is more noise. It can handle movements better, that's why.The suggested Network and the models made so far do not handle space and time similarly. A lot of the time, networks that use 3D convolutions can hide movies. It's expensive to add these changes and twists, and they might not get rid of everything that the movie doesn't need to know or say again. What about the Network that was suggested? You can position, focus, and get rid of fuzz with it. The dynamic filter also helps you make changes that will only work in one frame. Because of how they are built, this might not be possible with some types.

3.3. Attention

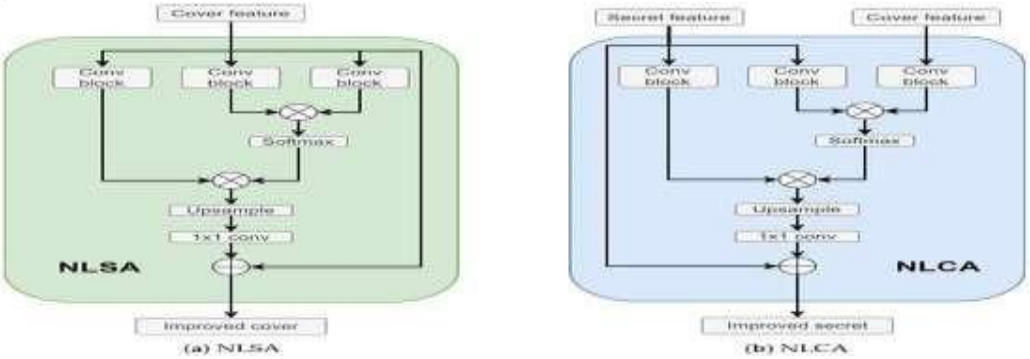
Focussing lets us work on different parts of a picture in a way that makes them stand out in computer vision. In a normal convolutional neural network (CNN) feature map, the data is often shown more than once in different channels. This is because photos are shared on the web. In this kind of CNN, every spot on the feature map is given the same weight. But traits may be more or less important at different times and places. Not every trait is important.

3.3.1. Attention That Is Not Local

Because of how attention works, Stego-STFAN can change how people watch movies and hide messages in them. People who sneak things into films are a good example of how important edges and fast pixels are. In other words, it's harder to keep a secret in these places without getting caught. Because that's how you think, the above method was changed by adding the NLSA block for non-local self-attention. The Network learnt more about the background, edges, and pictures that move quickly after this change. I also changed it so it now has the important parts of the cover and the secret movie's time-space story. In other words, it shows you the tougher parts of the secret movie. They can now pay more attention to those parts on the Grid. For the second task to be completed, the program would need to be changed to add the secret. You can learn more from Figure 4b.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 12s,2025
https://theaspd.com/index.php

Figure 4. Non-Local Attention.



The NLSA method comprises three main parts, as shown in Figure 4. First, the information is handled by three blocks still there. Only one of these blocks is in charge of the cover for it. On the other blocks, only secret films are shown. The next step is to use a matrix to mix two outputs from the blocks that came before it. The results are then turned on with a softmax. You multiply outputs to get it to work with NLCA. The last step is multiplying the softmax method's result by the result of the last block. There is an upsample because steps 2 are still in the extra blocks. Lastly, a 1 × 1 convolution layer is added. The attention block's input is mixed with the output of this layer.

3.3.2. Convolutional Block Attention Module is referred to as CBAM.

I added CBAM to our Network, as shown in Figure 5. This makes finding the most important information about the filters used in convolutional blocks in terms of space is simple. We can figure out what information is important between the frames and which parts of the channels are more linked by going through time through the channels. The Network can now easily check the link between frames.

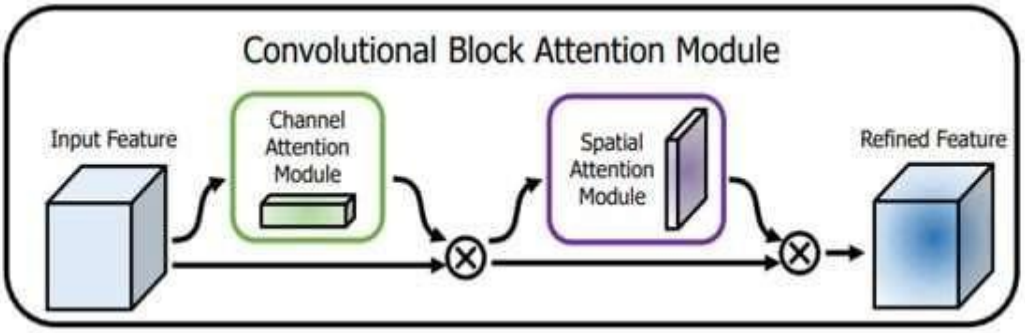


Figure 5. CBAM.

Following the steps in Figure 5, you should focus on channel and space in a certain order. Now, each branch will know "what" and "where" to focus on in the channel and space directions. Our module can help the Network's information flow by working out what data should be shown in the foreground and what data should be shown in the background. Next, research was done to find a way to measure how well the model could now be understood. In other words, using CBAM might help you get better results and understand them better at the same time. After looking at everything, it is believed that the better performance is due to less noise from unneeded waste and a better ability to focus on what is important. The last thing we do is ensure that the fact that CBAM is better at identifying objects in datasets shows that it can be used in many cases. When the part is meant to be light, the cost of input and processing is usually low. It works like this most of the time. But it's important to remember that CBAM's concept hasn't changed.

3.4. Group Normalization Block

Group normalization was used to make the value even and keep the numbers from getting too far out of range. Everything went as planned with this. This block normalization is not the same as batch normalization (BN). It looks at the single sample split into parts (groups) to figure out what to do next. After that, BN keeps the batch size straight. It's hard to train when there aren't many groups, just like in

BN, where the number of batches is very important. This is the case because they can only be used with a lot of people. Group Normalization kept the frame rates fixed, even when FAC breaks happened. We chose four different groups to show this.

3.5. DWT

You must frame a movie very well so that no one can find the message inside. Haar-DWT can help you find the good parts of the frame when handling the Picture and ensure they get extra care. Adding all the subbands to the second-order separation built on the work of and made it possible. It's still possible to hide the cover picture, so only the most important parts of the secret are known. As things stand, the stego-image was sent to the STFAN layer. The shell and the secret DWT were put together to make it. This was done so the Network could determine what frame parts and materials needed to be marked as hidden. We look into the best places to hide things in movies by modeling how areas are linked in steganography. We can see which secret spots get the most attention this way. It is considered that some people have trouble seeing parts of the movie. That's why this NLSA block was added: to draw more attention to hidden parts. Covering frames were used to see how space and time were connected. It was this block's job to find good hiding spots for service.Many different reference frames are used in this method to fake the link between two places in space and time. For the pixels around the action, there is a way to change their numbers. This is clear from the film still on the right lid and the container's frame after the process. It was easy to see the Picture of the secret frames with the help of an NLCA block. People found ways to connect the top and secret frames to make these pictures. It was important to watch the secret Picture's most important parts. Two kinds of attention were shown: nodal self-attention (NLCA) and nodal co-attention (1). However, they were put through STFAN instead of the other steps in this list. It was done since it's simple and doesn't cost much. Plus, it helps you stand out. It needs more space, so it has to go through STFAN. You can get a different view of the building with DWT. This is also good for the Network and how everything works together. WaveNet and compression networks are two deep learning methods already used with DWT to do similar work, mostly for hiding data. Putting the secret do it after all the work that was done. In this study, the same thing was done. Each frame was changed from RGB to YUV instead of going through a discrete wavelet transform (DWT). After that, the Y-shaped part was picked as the best place to put it.

3.6. Loss Function

It is important for the progress of a good video steganography technology that both the stego-image and the secret found are very good. Because of this, these two Network outputs need to be considered by the loss function. We get the mean squared error (MSE) from both the stego-image and the secret Picture like before. After that, it gives the secret that was found a weight of □. To keep things simple, the same number (0.75), which was used in, was also used here.

$$L_e = MSE(Cover - Container) \quad (1)$$

$$L_S = MSE(Srcret - Restored\ Secret) \quad (2)$$

Also, the model shouldn't make a loss that favors one loss over the other. That's why a loss was added to help the two sets come together and keep this difference in check:

$$L = Abs(L - L).Abs(PSNR - PSNR). \quad (3)$$

1

$$d \quad e \quad s \quad c \quad S \quad \alpha$$

where **Ω** is a hyperparameter that was set to 3. The punishment for PSNR difference between cover and secret is not as harsh if the number is too high. The number must be too low for them to learn how to work together instead of against each other. In the end, this amounts to the following losses:

$$L_{loss} = L_c + \beta . L_S + L_a \quad (4)$$

3.7. Final Considerations

Python was used to make it, and Google Colab was used to train it. The L4 GPU (22 GB) and 50 GB of RAM were used to train. This is the Convolutional block. It has a Leaky ReLU value of 0.01, a 3x3 convolution stride value of 1, and another 3x3 convolution stride value of 1. Then there should be another Leaky ReLU (with a value of 0.01), and this time the input should be added to the most recent convolution result. Every shot was 144 x 180 pixels and there were 5700 of them. The files UCF-101 and Weizmann Action are where they came from. Only 20% of these frames were used for tests, while 80% for lessons. Not even 10% of the 60 epochs meant for training were. It was set to 5 levels of patience. How much people learned at each step of the plan changed. A noise feed (0,0.015) was also used to even things out and show how things might change in real life. The person first saw the fake code.

The Stego STFAN Training Algorithm is the first algorithm.

Set up the factors
Hide the network by setting hiding_ network = STFAN_ hiding () // Start the hiding network
Setting up the restoring network: restoring_ network = STFAN_ restoring ()
Five points for patience if you want to stop early.
Setting best_ loss to 0 sets it to an initial value of infinity.
for every time d
Set epoch_ loss to 0 to start the process of epoch loss
Update the learning rate planner with l r_ scheduler. Step ()

there was no total loss for every group, so ▹ Set the total cost for the batch to zero.
for every frame
This is my_ d f: Container, Restored_ Secret, HHt_-1, RHt_-1. [at the moment]
Container and HH t are hidden by Network (HR t, HBt_-1, HB t, HHt_-1)
Secret restored, RH t_ r = network restored (RR t, RBt_-1, R t, RHt_-1)
The present frame in my_ d f is [Container, Restored_ Secret, HH t, RH].
Find loss for co by setting loss_ c = MSE (Cover, Container)
The MSE of Secret and Restored_ Secret is loss_ s. Find the loss for the secret
loss_ a = -abs (loss_ c - loss_ s) Find the exact difference loss
loss = loss_ c + δ·
Find the total loss by adding loss_ s and loss_ a.
Figure out the total loss by adding up all the losses.
end for
This is the total loss. b a c k ward () Back propagate total loss
The optimiser. Step (): Change the model's settings.
You can add up the time loss to get the total loss.
If epoch_ loss is less than best_ loss, the
Best loss = epoch_ loss Update best loss wait = 5 Again, be patient.
something else
Reduce patience by 1
end if
Let's say patience is 0.
Break † Time to stop early if you can't wait any longeZ
end if

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 12s,2025
https://theaspd.com/index.php

end for
Finish the process. Have you done anything else?

4. EXPERIMENTAL RESULTS AND ANALYSIS

The MATLAB R2016a tool was used with the given method. Many checks were done to ensure the stamp wouldn't show up or fade over time. A lot of the pictures showed what happens in drugs. The coolest new thing is a brand picture that can change sizes on the spot. This makes sure everyone looks the same. This project is run by a bunch of groups of 512x512 greyscale pictures. Another black-and-white picture of the same size that isn't from LL4. It could be a card. Both the file and our mark are LL4 sizes. We care about how big the host picture is. Our host picture is 512 pixels wide and 512 pixels tall (Li, G. et al 2023). When we finish LL4, our signature picture will be 64 bits by 64 bits. It's easy to put our Picture on new things that are the same. The host picture and the tag picture can never be the same size. How well did the watermarking method work? The NC, PSNR, or SSIM might help you find the pitch. The SSIM was also used to see how much the patterns were alike. Figure 6 shows the new pictures. Picture (a) shows the guests, and Picture (k) lists their names.The pictures (a–c) are all different. Pictures (d–e) are old background pictures from USC and SIPI. You can find Picture F on the computer. You could also do this: X-rays of the lungs were used to make pictures (g) and (h). There is an ECG beating classification collection and asthma is shown. There's something good about every shot here. Now that you know about these fish and dirty water, you can take shots of sea life. These are shown in (i) and (j). They were made from a fish picture.

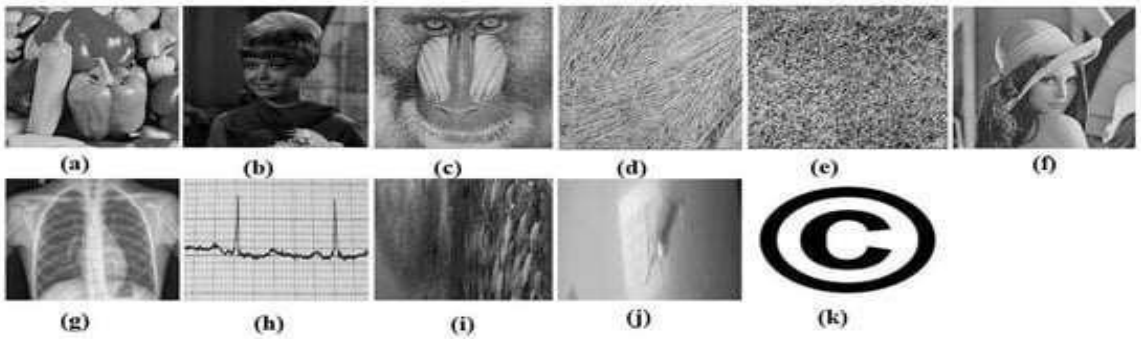


Figure 6. (a) Pepper; (b) Female; (c) Baboon; (d) Straw; (e) Grass; (f) Lena; (g) Chest X-ray; (h) ECG; (i) Fish Species; (j) Marine Animal; (k) Copyright (watermark image) are the input pictures.

So, this part looks at how the system can be strong, safe, and secure while being quiet. At the end of this part, we'll compare our suggested method to the already used to see how easy and useful it is.

4.1. Imperceptibility Analysis

We did a study (Table 1) to see how well our suggested method works with different sets of pictures. PSNR (dB) and SSIM were used to measure this. The PSNR number for each picture group in this table is over chest X-ray pictures had the highest PSNR level, at 48.9688 dB. This is what we saw. Because of this, we can say that this way makes it harder to find different kinds of pictures. Pics with and without watermarks can be seen in Figure 7.

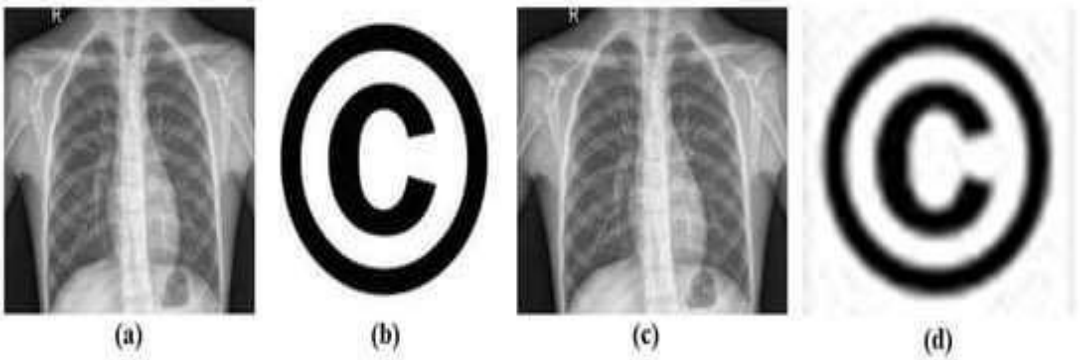


Figure 7. (a) Chest X-ray (Host Image); (b) Copyright (Watermark Image); (c) Watermarked Image; (d) Extracted Watermark.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 12s,2025
https://theaspd.com/index.php

Table 1. Imperceptibility Analysis for Different Groups of Images by PSNR (dB) and SSIM.

Type of Image	Picture Name	PSNR (dB)	SSIM
Miscellaneous	Pepper	48.9192	9997
	Female	48.6165	9995
	Baboon	48.7890	9999
	Lena	48.6339	9998
Texture	Straw	48.8687	9999
	Grass	48.8001	1.0000

Type of Image	Picture Name	PSNR (dB)	SSIM
Medical Image	Chest X-ray	48.9688	9997
	ECG signal	48.5490	9998
Underwater Image	Fish Species	48.8687	9997
	Marine Animal	48.5780	9993

4.2. Robustness Analysis

Noise, filter, geometric, picture blur, and mix attacks were tested on different sets of shots. Each of these shots has an NC number bigger than 0.9, which we found. In other words, our way can easily get back the stamp image when used on these pictures.

4.2.1. Noise Attack

Table 2 shows Speckle noise (SN), Gaussian noise (GN), Speckle Poisson noise (SPN), and Poisson noise (PN). That is the order. Because of this PN attack, the chest X-ray doesn't show what it is. The watermarked Picture's PSNR of 26.9433 dB is too low to see these things. However, we can still get rid of the labelled Picture with our way since NC is so close to 1. Figure 8 displays the PSNR (dB) and NC values for the chest X-ray medical picture after a successful noise attack.

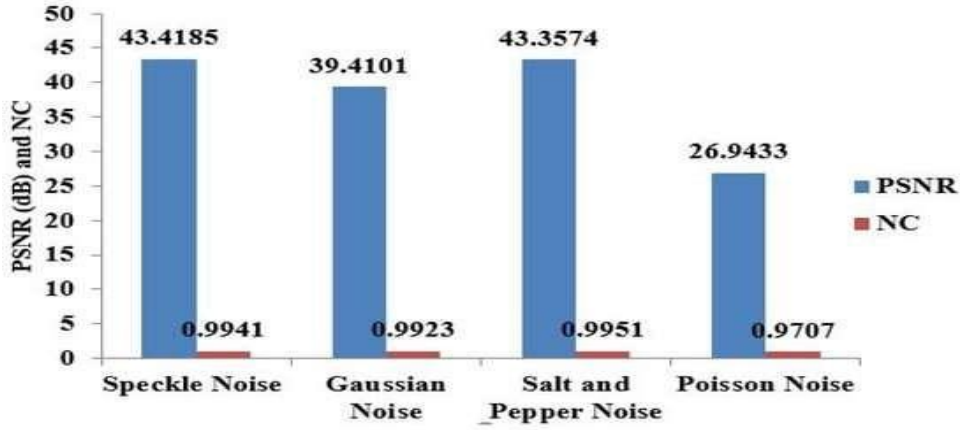


Figure 8. PSNR(dB) and NC values after noise attacks for chest X-ray image.

Table 2. Robustness Analysis for Noise Attack with Different Groups of Images by NC.

Type of Image	Picture Name	SN	GN	SPN	PN
Other Things	Pepper	9916	9911	9915	9469
	ffemale	9905	9909	9906	9313
	A baboon	9942	9924	9944	9446

Type of Image	Picture Name	SN	GN	SPN	PN
	Lena	9933	9920	9929	9674
Feel it	Straw	9912	9922	9936	9643
	Grass	9946	9955	9954	9776
Medical Picture	X-ray of the chest	9941	9923	9951	9707
	ECG reading	9957	9927	9953	9640
Picture of Underwater	Types of Fish	9944	9920	9944	9744
	Animal in the sea	9904	9908	9914	9468

4.2.2. Filter Attack

Table 3 shows that the median filter (MF), the Wiener filter (WF), and the low pass Gaussian filter (LPGF) all did different amounts of work. You can change how a picture looks with each of these hits. On Figure 8's chest X-ray, the PSNR reading was 35.3040 dB, the lowest since the AF attack. But the NC number is 0.9627, which shows that the method can remove the stamp on a picture.

Figure 9. PSNR(dB) and NC values after filter attacks for chest X-ray image.

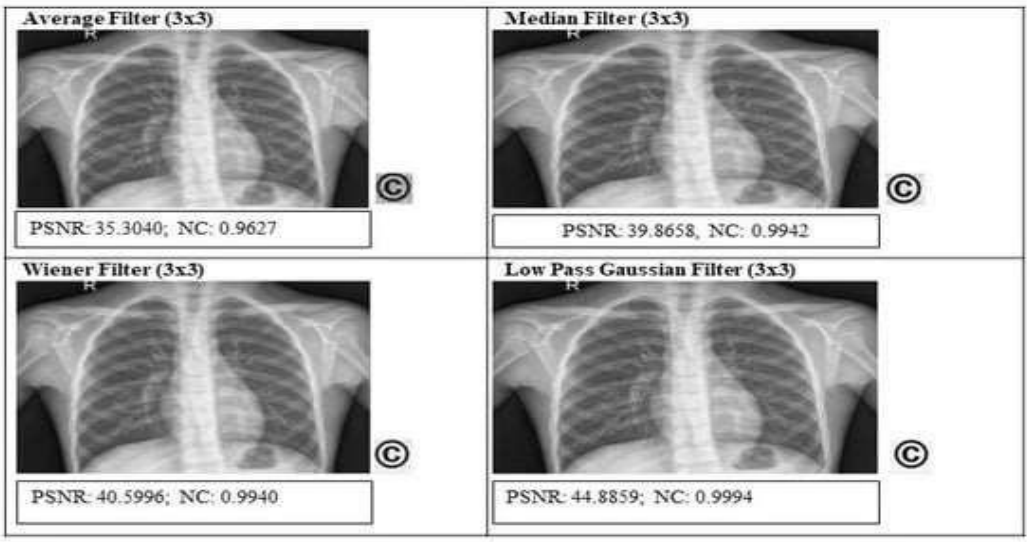


Table 3. Robustness Analysis for Filter Attack with Different Groups of Images by NC.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 12s,2025
https://theaspd.com/index.php

Type of Image	Picture Name	AF	MF	WF	LPGF
Miscellaneous	Pepper	9431	9789	9897	9978
	Female	9576	9978	9952	9993
	Baboon	9512	8753	9652	9963
	Lena	9713	9803	9936	9983
Texture	Straw	9641	9520	9732	9972

Type of Image	Picture Name	AF	MF	WF	LPGF
Medical Image	Grass	9245	9854	8967	9963
	Chest X-ray	9627	9942	9940	9994
	ECG signal	9233	8437	9517	9954
Underwater	Fish Species	9849	9658	9942	9989
	Marine	9939	9997	9993	9999

4.2.3. Geometric and Blur Attacks

Table 4 shows people talk about noise in pictures and geometry techniques like JPEG compression and shrinking. We used Matlab's "imresize" method to change the size of the Picture. The "imresize" method was used since B = imresize. This tool changes the size of picture A so that picture B is the right size. This is done with Matlab's built-in cubic approximation method. Every row and column in drawing A is a number. Figure 10 shows the PSNR and NC values for ways to change the size, shape, and blur of pictures are not shown. Instead, these numbers are shown.

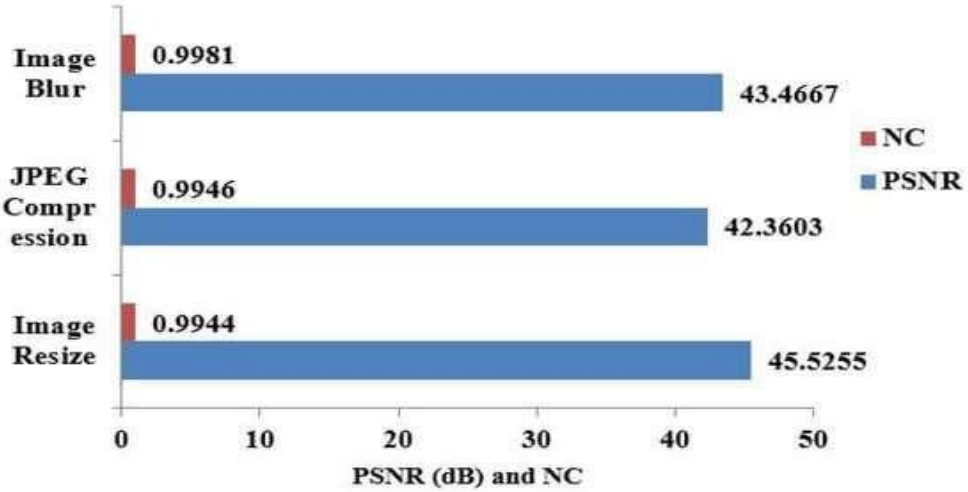


Figure 10. PSNR(dB) and NC values for a chest X-ray picture after geometric and blur assaults.

Table 4. Robustness Analysis of Blur Attacks with Various Image Groups by NC and Geometric (resize and jpeg compression) Attacks.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 12s,2025
https://theaspd.com/index.php

Type of Image	Picture Name	Size up	How to Compress JPEG	Picture Blur
Miscellaneous	Pepper	9934	9923	9974
	female	9907	9903	9978
Texture	Baboon	9948	9961	9970
	Lena	9935	9943	9988
	Straw	9936	9939	9978
	Grass	9943	9951	9969

Type of Image	Picture Name	Size up	How to Compress JPEG	Picture Blur
Medical	Chest X-ray	9944	9946	9981
	ECG signal	9929	9955	9952
Underwater	Fish Species	9950	9950	9991
	Marine	9917	9921	9995

4.2.4. Hybrid Attacks

You can see the results of the following combination hits in Table 5: MF+blur, SPN+WF, WF+GN, SN+MF, and MF+PN. Figure 11 shows that the PSNR for the chest X-ray was 26.7531 dB for the MF+PN attack as a whole. The NC number is so close to 1 though that the method can still get the logo picture right.

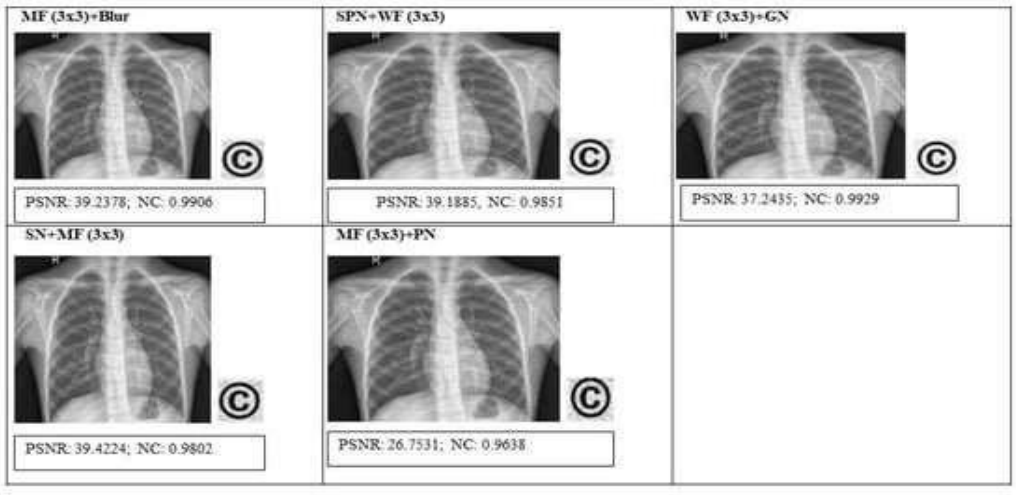


Figure 11. PSNR(dB) and NC values for a chest X-ray picture after hybrid assaults.

Table 5. NC's Robustness Analysis of Hybrid Attacks Using Various Image Groups.

Type of Image	Picture Name	MF+Blur	SPN+WF	WF+GN	SN+MF	MF+PN

Other Things	Pepper	9656	9834	9885	9750	9623
	A woman	9952	9841	9934	9865	9824
	A baboon	8657	9549	9654	8772	8907
	Lena	9718	9805	9902	9730	9554
Feel it	Straw	9412	9657	9755	9469	9371
	Grass	9834	8602	8940	9840	9820
Medical Picture	X-ray of the chest	9906	9851	9929	9802	9638
	ECG reading	8329	9438	9493	8335	8859
Picture of Underwater	Types of Fish	9589	9907	9919	9627	9417

Type of Image	Picture Name	MF+Blur	SPN+WF	WF+GN	SN+MF	MF+PN
	Animal in the sea	9995	9910	9979	9905	9269

4.3. Security Analysis

An Arnold map was used to keep the card picture safe. Both people who send and receive the Arnold map hide the iteration number, also called the key. It's very safe and private now. You can't get the stamp picture if the key is missing. Figure 12a shows how important it is to press the right key to get the stamp picture out. You will never be able to get back the right title picture for Figure 12b, no matter what else is true. Picture pieces on a fake stamp are close to each other but don't fit together to make a picture. This means the Picture doesn't show anything. Our plan is safe enough for everyone to agree with it.

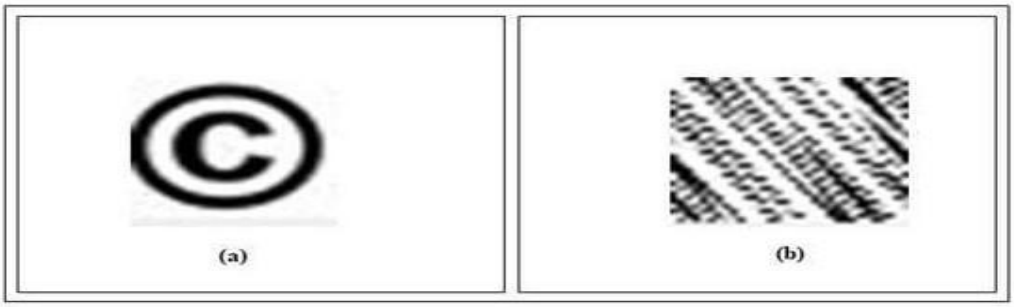


Figure 12. Extracted watermark; (a) with right key 32; (b) with wrong key.

4.4. Capacity Analysis

When the bit numbers are raised in the case shown, PSNR and SSIM stay the same. The picture signal-to-noise ratio (PSNR) compares an image to its source to see how good it is. It sounds better when the PSNR number is higher. A number above 30 is good, according to most people. The PSNR for the pepper picture in Table 6 is 47.5334 dB. Many things about the watermarked Picture are the same as the source picture. In other words, the Picture is very good.

Table 6. PSNR(dB) and SSIM capacity analysis.

Host Picture	Picture Watermark	Bit Size	PSNR (dB)	SSIM
Pepper	Don't Copy	1024, 512, 256, 128, 64, 32, 16	47.5334	9984
		8	48.8636	9988
		2, 4	Infinite	1

Another way to find out how similar two pictures are across their structures is to use the SSIM measure. From -1 to -1, SSIM values range from 1 to 1. A value of 1 means there is an exact match. SSIM gave a result of 0.9984, which is very close to 1. This means that the watermarked Picture and the Picture there before look a lot alike.

4.5. Comparison with Existing Methods

There are parts of our plan that are the same as things that are already done. These links are shown in Table 7. The method, the cover image type, the cover image size and colour, the watermark image type, the watermark image size and colour, PSNR (dB) and SSIM, security methods, and the program were all used. What our suggested method is good at can be seen in these ways. They are used to compare its features to those of other methods that are already in use. As shown in this table, the PSNR (dB) value of the watermarked Picture made the way we suggested is higher than the PSNR values made the way things are now. The SSIM we have now is higher than any other way, though. Now, we'll compare our suggested mixed watermarking method to other ones to see how well it works and how hard it is to spot.

Table 7. Comparison of the Proposed Method's Features with Newer Approaches.

Method	DCT + DWT + SVD	DFT + SVD	DWT + DCT + SVD	DWT + DCT	Fourier transform+ QR decomposition	NSCT + RDWT	DWT + SVD		NSCT + DWT + SVD	DWT + SVD
Cover Image Type	Lena	Lena	Medical image	Pepper	Lena, Baboon, House, Sailboat	Pepper, medical image	Lena		Lena	Pepper, Lena, Chest X-ray
Cover Image Size and Color	512 × 512, grayscale	512 × 512, grayscale	1024 × 1024, grayscale	512 × 512, grayscale	512 × 512, color	512 × 512, grayscale	512 × 512, grayscale and color		512 × 512, grayscale	512 × 512, grayscale
Watermark Image Type	Panda	Panda	Hospital logo. text watermark	Text image	Pepper	Logo	Logo		Cameram an	

Watermark Image Size and Color	64 × 64, grayscale	64 × 64, grayscale	32 × 32, 128 × 8 grayscale	64 × 64, grayscale	512 × 512, color	32 × 128, grayscale	256 × 256, grayscale	32 × 32, binary

4.5.1. Imperceptibility Comparison

The PSNR (dB) and SSIM of our suggested method are shown in Figure 13. It is compared to other methods. The Picture shows both of these things. Look at Figure 13a to see the PSNR values for chest X-rays taken with our suggested method and other popular ways of marking. You can get the same PSNR number (48.97 dB) in other ways, but ours is the best. This Picture of Lena has an SSIM made the way we suggested next to SSIMs made with other marking methods. This can be seen in Figure 13b. There is a bigger SSIM number for the way we suggested (0.9998) in this graph than for the other options.

Figure 13. (a) PSNR values of various methods; (b) SSIM values of various methods.

Because of this, our suggested method promises a higher level of imperceptibility (high PSNR and SSIM) compared to the more recent ones.

4.5.2. Robustness Comparison

Now there are many hard-to-understand ways to handle noise, filter, and blur attacks on chest X-ray images, but the way we did it worked great. Figure 14 shows that GN and SN were not the only ones who hit. Figure 14a shows how to hit the SN with the highest NC number, 0.9941. The largest NC number compared to the current GN attack methods can be seen in Figure 13b. It is 0.9923. The SPN attack has the highest NC value of all the new methods used, as shown in Table 8. Figure 15 shows our test to see how MF and LPGF attacks stack against each other. Look at 15a. Our MF attack method has the best NC value (0.9942) compared to other methods. The difference between our method (0.9994) and the other one (0.9996) shows that our method is better at stopping the LPGF attack shown in Figure 15b. For the picture blur attack, our way worked better (0.9981) than the other way, which worked 0.9780 better.

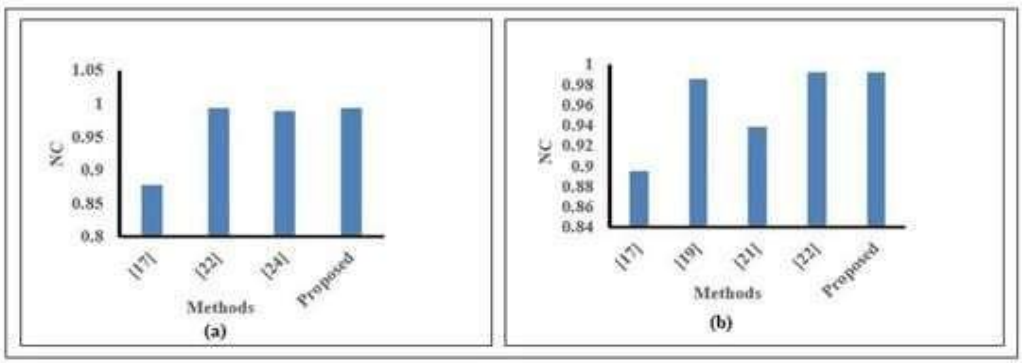


Figure 14. Robustness comparison for SN attack (a) and GN attack (b).

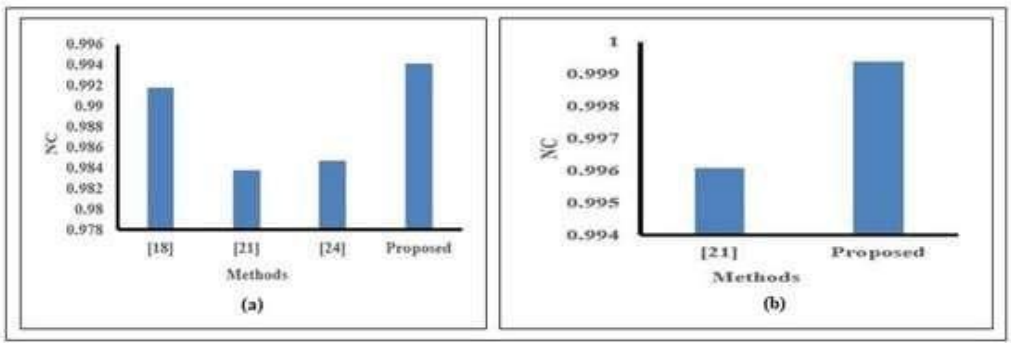


Figure 15. Robustness comparison (a) for MF assault and (b) for LPGF attack.

5. Conclusions

Some people said that DWT and SVD could be used together to make a strong sign that no one can see. We test the suggested method with several different host shots and compare the results to those of other, more current methods that have been written about. The clear stamp on a picture will tell you if it's true or not. The ones that are already in use are clearer than these. Something needs to be turned off at 48.9688 dB. Also, this way is better than those used now at blocking noise, filters, and picture fuzz. The NC number is often the same as or better than what other tools are currently using. A lot of noise hits use this number. We should use the NC number 0.9876 as a filter attack, but we're not. When we used the picture blur attack, the NC value was 0.9981, higher than any other way. The Arnold map makes the system safer, which makes it safer. With our method, you can use either of the cards with an SSIM number of 1. You can't remove the tag picture from DWT because of how shift variation works.You can't change the background or host picture as much either. We'll use more than one DWT to determine how to improve things this time. Don't forget that the FPD issue is not with this job but how it was shown. Removal, genetic, and raw force should all be used the next time we work on this subject. Besides that, our method works for stamp pictures of any size or style. This is how we can tell how strong the system is. We will use deep learning and guess in some of the things we do. The Picture will be cut, turned, and shrunk by geography hits. The Picture will always look good with blind watermarking. After that, the mark is taken out of the Picture correctly. Finding that first Picture isn't always easy or helpful. This is important to remember as you work on the tool that lets you add watermarks without seeing them. You won't be able to find them again because their names will have changed. People are still discussing ways to keep the Internet of Things (IoT) safe. A generative adversarial network is another way to name pictures with less noise. When you label a picture, you can find fake digital photos because different photos act in different ways.There are "state-of-the-art" ways to find fake accounts on every Network. The best way to do it is with the picture response non-uniformity method. Since fake sites don't have watermarks, they are easy to spot. Next time, we can use watermarks, generative adversarial networks, or picture replies that don't always look the same to make it safer. This person already knows these things. You can also stop people who shouldn't see certain pictures from seeing them with ending code. You can also hide info in several ways. If you start with the feature extraction method, it does work. After that, we'll add a way to end code blocks and security steps that change based on whether the IoT or seen link is used. Now things will get better. If we use the same Arnold mapping method to remove it as we did to add it, we might be unable to get back to the Picture before the name. This shows that you can change the name on the Picture. You can now get the stamp strength alpha number in several different ways. The first one is a good size.

Declarations:

All authors declare that they have no conflicts of interest.

REFERENCES

1. Ahmadi, M.; Norouzi, A.; Karimi, N.; Samavi, S.; Emami, A. Redmark: Framework for Residual Diffusion Watermarking Based on Deep Networks. Expert Syst. Appl. 2020, 146, 113157.
2. Baluja, S. Hiding Images in Plain Sight: Deep Steganography. In Neural Information Processing Systems (NIPS); Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 2069–2079.
3. Barni, M.; Bartolini, F.; Piva, A. Improved wavelet-based watermarking through pixel-wise masking. IEEE Trans. Image Process. 2001, 10, 783–791.
4. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M. Generative adversarial networks. arXiv 2014, arXiv:1406.2661.
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26–30 June 2016; IEEE: New York, NY, USA, 2016; pp. 770–778.
6. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
7. Holub, V.; Fridrich, J. Designing Steganographic Distortion Using Directional Filters. In Proceedings of the 2012 IEEE International Workshop on Information Forensics and Security (WIFS), Costa Adeje, Spain, 2–5 December 2012; IEEE: New York, NY, USA, 2012.
8. Holub, V.; Fridrich, J.; Denemark, T. Universal Distortion Function for Steganography in an Arbitrary Domain. EURASIP J.

Multimed. Inf. Secur. 2014, 2014, 1.
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Neural Information Processing Systems 25 (NIPS); Curran Associates Inc.: Red Hook, NY, USA, 2012; pp. 1097–1105.
10. Li, B.; Wang, M.; Huang, J.; Li, X. A New Cost Function for Spatial Image Steganography. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; IEEE: New York, NY, USA, 2014; pp. 4206–4210.
11. Li, F.; Yu, Z.; Qin, C. GAN-based spatial image steganography with cross feedback mechanism. Signal Process. 2022, 190, 108341.
12. Li, G., Li, S., Li, M., Zhang, X., & Qian, Z. (2023, June). Steganography of steganographic networks. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 37, No. 4, pp. 5178-5186).
13. Liu, J.; Ke, Y.; Zhang, Z.; Lei, Y.; Li, J.; Zhang, M.; Yang, X. Recent Advances of Image Steganography with Generative Adversarial Networks. IEEE Access 2020, 8, 60575–60597.
14. Luo, X.; Li, Y.; Chang, H. DVMark: A Deep Multiscale Framework for Video Watermarking. arXiv 2021, arXiv:2104.12734.
15. Luo, Z., Li, S., Li, G., Qian, Z., & Zhang, X. (2023, October). Securing Fixed Neural Network Steganography. In Proceedings of the 31st ACM International Conference on Multimedia (pp. 7943-7951).
16. Marvel, L.M.; Boncelet, C.G.; Retter, C.T. Methodology of Spread-Spectrum Image Steganography; Army Research Lab: Aberdeen Proving Ground, MD, USA, 1998.
17. Ruanaidh, J.; Dowling, W.; Boland, F. Phase watermarking of digital images. In Proceedings of the 3rd IEEE International Conference on Image Processing, Lausanne, Switzerland, 19 September 1996; Volume 3, pp. 239–242.
18. Shrestha, A.; Timalsina, A. Color Image Steganography Technique Using Daubechies Discrete Wavelet Transform. In Proceedings of the 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Kathmandu, Nepal, 15–17 December 2015; IEEE: New York, NY, USA, 2015; pp. 1–7.
19. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
20. Su P-C, Cheng Y-H, Kuo T-Y. JSN: Design and Analysis of JPEG Steganography Network. Electronics. 2024; 13(23):4821. https://doi.org/10.3390/electronics13234821
21. Subramanian, N.; Cheheb, I.; Elharrouss, O.; Al-Maadeed, S.; Bouridane, A. End-to-End Image Steganography Using DeepConvolutional Autoencoders. IEEE Access 2021,9, 135585–135593.
22. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; IEEE: New York, NY, USA, 2015; pp. 1–9.
23. Vergara, G. F., Giacomelli, P., Serrano, A. L. M., Mendonça, F. L. L. D., Rodrigues, G. A. P., Bispo, G. D., ... & Sousa Júnior, R. T. D. (2024). Stego-STFAN: A Novel Neural Network for Video Steganography. Computers, 13(7), 180.
24. Wang, H.J.M.; Su, P.C.; Kuo, C.C.J. Wavelet-based digital image watermarking. Opt. Express 1998, 3, 491–496.
25. Weng, X.; Li, Y.; Chi, L.; Mu, Y. Convolutional Video Steganography with Temporal Residual Modeling. arXiv2018, ar Xiv:1806.02941
26. Weng, Xinyu, et al. "High-capacity convolutional video steganography with temporal residual modeling." Proceedings of the 2019 on international conference on multimedia retrieval. 2019.
27. Westfeld, A. F5—A Steganographic Algorithm. In Proceedings of the Information Hiding; Moskowitz, I.S., Ed.; Springer: Berlin/Heidelberg, Germany, 2001; pp. 289–302.
28. Wiegand, T.; Sullivan, G.J.; Bjontegaard, G.; Luthra, A. Overview of the h. 264/avc video coding standard. IEEE Trans.
29. CircuitsSyst. Video Technol. 2003,13, 560–576.
30. Xu, S., Li, Z., Zhang, Z., & Liu, J. (2022). An End-to-End Robust Video Steganography Model Based on a Multi-Scale Neural Network. Electronics, 11(24), 4102.
31. Xu, Shutong, et al. "An End-to-End Robust Video Steganography Model Based on a Multi-Scale Neural Network." Electronics 11.24 (2022): 4102.
32. Yu, C.; Cheng, S.; Zhang, X.; Zhang, X.; Tang, Z. Reversible Data Hiding in Shared JPEG Images. ACM Trans. Multimed.
33. Comput. Commun. Appl. 2024, 20, 1–24.
34. Zhu, J.; Kaplan, R.; Johnson, J.; Fei-Fei, L. Hidden: Hiding data with deep networks. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.