Cryptography

AN APPROACH TO IMPLEMENT CRYPTOGRAPHY IN QUANTUM NEURAL NETWORKS AND DNN-BASED KEY HIDING

Kritika Chugh
SUID: 882046659

Hemanta Kumar Pattnaik
SUID: 770755090

Abstract

There are various ways to store keys – memorization of keys is one method. But do we want to rely on the memory of the people for this? May be not. There are other possible ways to store keys more secure without relying on the memory of the people.

Such as platform trusted and secure modules, Hardware Security Modules (HSM). There is a tradeoff between the choice. In this paper we are going to talk about various other famous cross concepts in which cryptography meets neural networks, very famously also called as Steganography. Steganography is basically a way to hide a message (any format) into another file (any format). The advantage lies in the process of cryptography using DNN (Deep Neural Network) as the message does not attract any sorts of attention at all during the transmission as it has now been disguised as another. This has a lot of legal benefits too in many areas where having an encrypted message comes with legalities. This technique is not just simple and discrete but also begs little complexity. In this paper we will be more closely inspecting key hiding ways inside the digital images.

Digital Images

Now, Digital images are Ubiquitous and they come in various formats (jpeg, png , etc.). We see computers putting together these images using pixels and each pixel carry a color assignment to them. So, we can consider it to be a grid organization of x and y with each grid carrying a color value. These color values follow the RGB scheme or Red-Green-Blue Scheme. Just to add more appropriate idea to it, there are other schemes that exist as well. But the idea here is that we can derive practically any color using these three colors. We can control the brightness of these pixels that range from 255 to 0, where 255 is the maximum bright and 0 the least respectively. So from this we can safely say that these pixels follow some sorts of matrix or table data structure and each image is layered representation of basically different brightness levels of Red layer, Green layer and Blue layer, where the brightness ranges between (0-255)

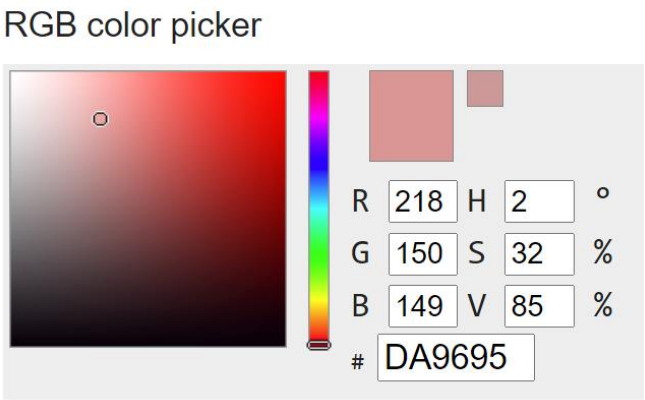


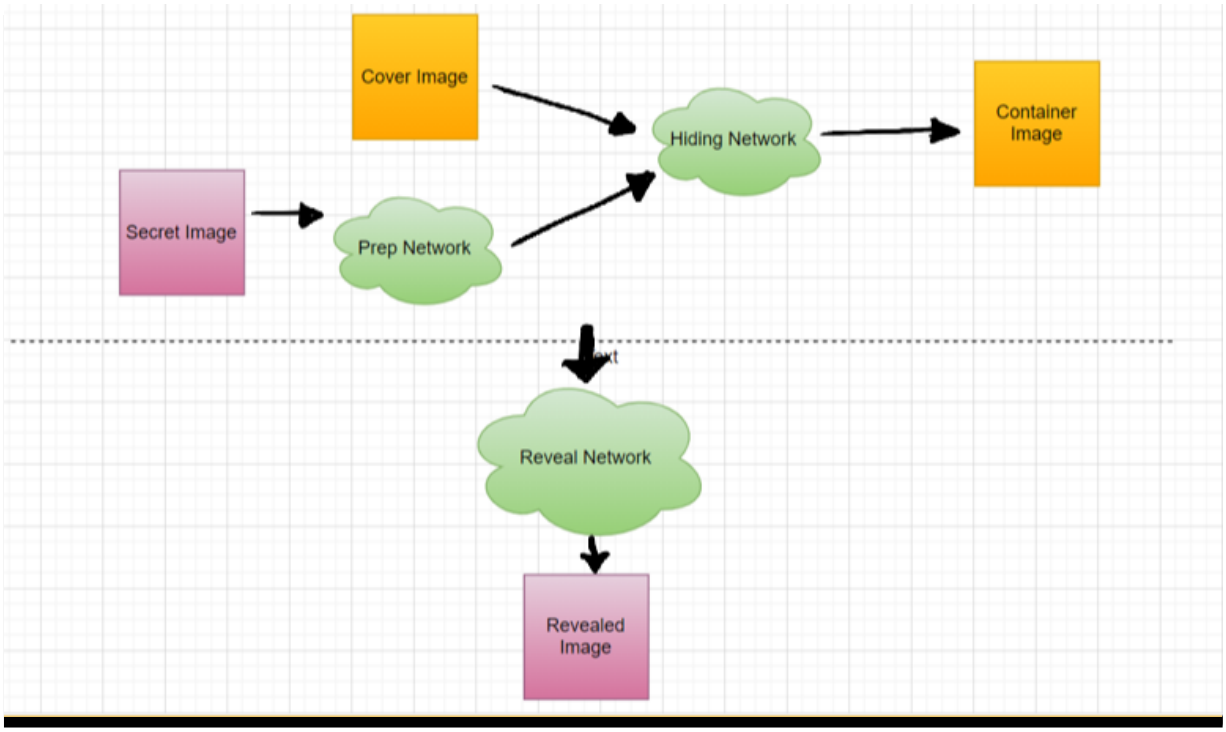
FIGURE 1 : DEMONSTRATING RGB COLOR ON DIFFERENT BRIGHTNESS LEVEL. REF: RAPID TABLES

Preparing the Images for disguising.

Let us prepare our images in question for this approach.

The Technique explained

We utilize the LSB or Low significant Bits of the image or the per slide of the video and try to altercate that region of it. We place the image to be hidden and its pixel and place them either uniformly or distributive. There are several other advance models exist already around it. We want to add our take on it using the shuffling techniques for the video. So, steganography can be applied. Our idea is that we can simply apply steganography using LSB method (Low significant bit method) frame by frame basis. But on top of that we can also associate a kind of shuffling that is not at all random. That shuffling will entail some sort of a shuffle algorithm which we will discuss in next section. So basically, we are not just implementing steganography on the images but also doing some sort of shuffling on the frames. The message makes sense if the frames are more aligned uniformly. We already have many examples of shuffling which is legally counted as forgery in surveillance systems. We can utilize this concept along with Steganography and can achieve great results.



The system learns or the DNN learns to compress the image or video or text from secret image to least suspected portion of the image. It is quite simple to understand that we want to basically blur somehow the pixels of the hidden image into the cover image without compromising the quality of the original image at all.

Prep Network -> Prepares the secret image and makes it ready to be hidden. It makes the size of the cover image and the secret image same (a pre-requisite)

The secret hidden DNN network -> we feed in the output from the preparation network along with the image (re-sized to match the cover image) and the cover image to this network.

Decoding Deep Neural Network -> It acts on the receiver end to decode the image it received. It basically acts a detective to reveal the image lying underneath the cover image.

One should always note that these NN (Neural Networks) are independent of the images. They are trained with less data but should be able to uncover results. The goal is to uncover images (any secret image from cover images) and operation should not be expensive. This brings us to another aspect of our architecture.

How is the network trained then?

There are 3 networks involved in the system and all 3 of them needs some sort of training without being expensive. So, we are not training them with huge data sets. Instead reducing the error during training is our goal.

Let us suppose, X and Y are the cover images and Secret images, respectively.

L = loss in the network.

HP – hyper parameter

$L(X, X', Y, Y') = ||X-X'|| + (HP)||Y-Y'||$

Since it is a learning model, we need a hyper parameter. For simplicity consider that if HP is too high then the learning may have high changes of collisions. And if it is too low then we will miss on important patterns from the data set.

Now our idea is to make sure that we do not simply add the hidden image to the LSB (least significant bit area of the cover image). We want extra protection. The idea is to somehow find the way of store the image in the LSB and other parts of the container image too without compromising the pixels and quality for the original image.

FIGURE 2 : IMAGE 1 IS THE COVER IMAGE, IMAGE 2 IS THE SECRET IMAGE , THE LAST IMAGE SHOWS HOW COVER IMAGE IS CONTAINING THE SECRET IMAGE. NOTICE THAT THE PIXELS ARE SOMEWHAT BROKEN AND CAN BE OBSERVED. SINCE WE HAVE UTILIZED THE INBUILT

We are attaching the code in the zip file.

Code and algorithm for RGB in grid implementation idea reference and citation [3] [3.1]

Quantum computing in cryptography and roles that help in Stenography techniques

So far, we discussed Stenography as an application to hide images. There also exists a relatively unexplored and expensive way to do the scaled version of stenography using Quantum computing too. We are also going to use the results presented by [1] paper. We are going to discuss LSB quantum cryptography and steganography.

Way back a mechanism was designed using quantum computing or key distribution using quantum computing so that if two people are sharing secret messages their keys are not compromised. This idea was basically a little different from general protections that exist in cryptography today. The idea was that if the eaves dropper is listening to the two people talking over the network then it will add noise to the communication channel of the two people sharing messages. And thus the 2 parties can throw the key bits they have and start again.

Another major reason cryptography using quantum computing became popular was because there is a factoring problem involved during the inversion. We usually see these inversions during the public key cryptography methods such as RSA and that is a problem difficult to solve in decent amount of time using classical computing power, we have available today. Although quantum computing can easily compute such problems fast.

Now there is another method famous called as Quantum Key Distribution and its secure for one and one reason only – It does not use math. It works using photons. And currently very secure. Both sender and receiver are supposed to have a general knowledge of which beam splitter to use in order to encrypt and decrypt the message from which arises a unique optical key.

So, in this technique of using QKD as stenography technique the message transmission is done and post that it is cluttered with false information for the eave's dropper.

Future Work possibilities and scope according to personal analysis.

Now, Stenography has given rise many other fields such as Steganalysis. This is a science where cryptography and Machine learning using Deep Neural Networks Overlap. Finding even if an image contains any hidden information from the pool of images the exist on the internet is like finding a needle in the puddle of haystack. We personally after referring to many research papers during this time have concluded that even though brute force steganography using statistical analysis can be used to determine with some accuracy that an image is a cover for the hidden image or message, but still this field is yet very naïve and need more work and performance efficient and accurate approaches.

Here the sender encodes the bits using either A = {|0> or |1>} or B= {|-> or |+>} energy levels only.

Where, $|+\rangle = 1\sqrt{2}\,(|0\rangle+|1\rangle)$ and $|-\rangle = 1\sqrt{2}\,(|0\rangle-|1\rangle)$. This is basically the bit representation in Quantum computers. The 2 users shares messages using this qubit format . the receiver would

receive all the qubits from the encoded message and would afterwards use randomization to measure those qubits on either the basis of A or B (where , A = {|0> or |1>} or B= {|-> or |+>}). The sender will announce over the deep neural network the bases it used for the encoding. The receiver will acknowledge which he used for the message. Notice that until now roughly 2n bits exist on the network. Now the sender will again share the first half of the bits she shared before and both the receiver would check for the error rate. If the error rate is more than the threshold then the communication for the remaining 2n bits is aborted and is initiated again. Notice that this technique uses error rate similar to the HP (Hyper parameter we used in the stenography technique above). The goal is to minimize the HP to a particular threshold and minimizing the loss function 'L' too.

Conclusions

So, we have so far seen the techniques that helped us to see how one can utilize the simplest of the concept of image pixel color and intensity adjustment to be able to hide the secret image using the cover image. Along with that we have also identified the possibility of quantum computing being able to play a big role in the field of stenography. We have though in our paper tried to stray away from discussing any thing more in detail about the neural networks so as to not to digress from the essence of cryptography and key hiding using DNN.

References

https://www-sciencedirect-com.libezproxy2.syr.edu/science/article/pii/S0375960115000997?via%3Dihub [1]

https://web.stanford.edu/class/cs101/image-1-introduction.html [2]

https://towardsdatascience.com/hiding-data-in-an-image-image-steganography-using-python-e491b68b1372 [3]

file:///C:/Users/kriti/Desktop/Hiding_an_Image_inside_another_Image_using_Variabl%20(1).pdf [3.1]

https://unthinking.photography/articles/hidden-in-plain-sight-the-steganographic-image

Tools and other references worth mentioning:

Color and hex code determining tool https://htmlcolors.com/google-color-picker

Evaluation of deep neural networks https://www.youtube.com/watch?v=HTu7RokJsxc

Google images for download of the 2 images in use.
https://www.google.com/search?q=google+images&oq=google+images&aqs=chrome..69i57j0l5j46j69i60.1820j0j4&sourceid=chrome&ie=UTF-8

Code reference: Kelvin Salton do Prado (see the how to run file)