Image Processing Encryption

Dhulfiqar Mohammed Saad SHABEEB¹, Saadaldeen Rashid Ahmed AHMED² Abdullahi Abdu Ibrahim³

¹Electrical and Computer Engineering, Altinbas University, Istanbul, Turkey
²Computer Science, University Of Tikrit, Iraq
³Electrical and Computer Engineering, Altinbas University

E-mail: dhulfiqarmohammed1996@gmail.com

Abstract

The aim of this research paper is to develop a new approach of image processing encryption using machine learning techniques in python so that the human cannot understand the images because it is in encrypted form and can be securely transfer to its destination. We also used the computer-generated Holography (CGH) technique to encrypt and decrypt images. We first implement an existing algorithm and verify the claims of the authors. We then investigate higher dimensional Baker maps for image encryption. For this, we first propose a new interpretation for the Baker map in terms of a path function S. We then apply the higher dimensional maps for image encryption and experimentally conclude that 3D Baker map suffices for encryption. That is, there is no perceptible performance gained when using higher dimensional Baker maps. Next, in an attempt to use chaotic maps for the diffusion mechanism in the encryption scheme, we embed the diffusion process into the confusion process. For this, we first propose an alternative view of a 2D image as a 3D structure using the binary representation of the image intensity values. We extend this scheme from grayscale images to color images and show its immense value in color image encryption. Lastly, we propose a Baker map based on random walk of the image. Here, we employ sparse decomposition of images as a method of generating the random paths. Random walk-based Baker maps would be more difficult to break than traditional Baker maps because of the chaotic behavior in the walk itself. The significance of images and their sharing is increasing day by day. Their security is becoming an important issue while transferring over a public network. To protect images from hacker’s secret sharing is one of the best techniques. The secret sharing is a way to share a secret with n participants and then setup is made for t or more number of participants who must contribute to revealing the secret. Here t ≤ n is known as a threshold which must be achieved for secret reconstruction.

Keywords: Machine learning, 3D Baker map, Secret image sharing, Computer Generated Holography (CGH).

1. INTRODUCTION

Holography is the art of presenting a 3D image on a 2D surface. Holograms are generated by recording the intensity and phase of a light [1]. A photograph only records the intensity of light, but a hologram records both the intensity and phase. For this reason, when it is printed on a holographic plate, the object appears to be three dimensional. A traditional hologram is generated by recording the light scattered from an object when the object is illuminated from a coherent laser source [2].
reference light from the same source is then taken and when both the light waves are superimposed on each other then it produces a holographic pattern. Which then if projected on a holographic plate a 3D visualization of the object is generated [3].

Fig 1: Recording of a Traditional Hologram.

A computer-generated hologram which is popularly known as CGH is a bit different from traditional holography. For a digital hologram the presence of the real object is not necessary, and the intensity of the coherent source is mathematically calculated [4]. The interference pattern forms the two waves, which are then calculated by algorithms and thus a final holographic pattern is generated. Holographic interference pattern can be used in image encryption. An image can be broken down and the spatial domain of the image can be converted into frequency domain using different algorithms [5]. If the far field pattern of the image is printed, then it cannot be decrypted by anyone as it is nothing other than some scattering pattern. If the far field amplitude scattered from the object and the reference light can be calculated, then the image can be perfectly reconstructed. In this case the reference wave can work as the encryption key in reconstruction of the image. Without the calculation of the reference light the image cannot be reconstructed. So, by calculating the far field amplitude of the reference beam it can be added with the far field amplitude of the light scattered from the object to obtain the final image. But as the object is not physically present here the scattered light from the object is calculated by using mathematical models. Deep Neural Networks are data and resource hungry; they require a good amount of system memory and multiple processing cores. In computers of daily use, these requirements are fulfilled by powerful dedicated GPUs. However, in SBCs the integrated GPUs share limited system memory with the CPU. When the system memory is exhausted, the CPU starts using secondary storage space as the virtual memory. The hard drives used as secondary storage are thousand times slower than the system memory, and as soon as the CPU starts using it, the performance hits the floor due to latency of disk read/write operations. Furthermore, the SBCs are made with top priorities being portability and small form factor. Small form factor makes SBCs suitable for DIY and IoT projects [6]. If an image encryption method with reliable accuracy can run on an SBC with satisfactory running time, it will be a huge breakthrough in the embedded industry. The problem of biometric human identification can be approached in several different ways. Analysis of fingerprints has historically been very successful and has the advantage of being highly accurate, but it suffers the disadvantage of being quite intrusive. A non-intrusive alternative to fingerprints, which has been shown to hold the same discriminative power. Looking to nature, it turns out that the ability to perform image recognition is so important for humans, that the human brain has evolved a center, the fusiform, specialized for this task [7]. This tells us two things, nature has deemed this ability advantageous enough to spend significant amounts of time and energy on acquiring it, and image recognition is a problem that probably needs specialized processing to be solved satisfactorily.

2. METHODOLOGY

We are proposing a method of encrypting and decrypting holographic interference pattern by using the principle of digital Holography using python as our programming language. For this firstly we will be creating our very own python library for generating computer generated holograms. The whole process will complete in four stages. Firstly, we will generate a holographic pattern and the reconstructed final image by using python and a computer vision library OpenCV. This will allow us to take any input image and convert the matrix into a holographic matrix which can later be reconstructed by our algorithm. Secondly, while reconstructing the image one of the major problems is that two images of opposite orientation will be formed, and they will overlap. So, the supplied image needs to be cropped and the dimensions of the images need to be supplied for the creation of the reference beam. So, because of this overlap in the final reconstruction we will not be able to see the exact image, rather we will see two images overlapping on each other. We are also suggesting a solution for all of these problems so that it will work on images on any size. We are providing a method with which the reconstructed image will not be overlapping. Our algorithm will automatically rescale the image and a reference beam will be calculated according to that size of the image. We have selected Discrete Fourier Transform (DFT) method for converting our image from a spatial domain into a frequency domain. We have chosen python as our programming language because it is free and open source.
Though most of the work in this field is done in MATLAB but MATLAB is closed and proprietary. So, at times it is not possible to dig deep into the methods and the algorithms. Python is fast and reliable with a huge library support. Python also has more than one library dealing with the Fast Fourier Transform (FFT) algorithm. We will be also using OpenCV and numpy libraries of python for processing our images and to calculate the FFT. We are suggesting a single key or symmetric cryptography technique for encryption of the image. For this we have to take the reference wave matrix and fill it with randomly created variables. For the holographic pattern we need a blank reference wave with point intensity because we are assuming that the light is coming from a coherent point source. So, if we will generate a reference wave with random pixels and we will note down the pixels and store them as our variables and we will take intensity variable which will be greater than any of the random variables we will take. we are suggesting a decryption method from the pattern we have generated and the key. Our decryption algorithm will first calculate the intensity and the pixel variables from the key.

2.1 Architecture of GCH

A holographic pattern can be obtained from an image using the computer generated. There are several steps in computing the interference pattern from the input image matrix. Several algorithms are then applied on the interference pattern to obtain the final reconstructed image. A flow chart of the working principles of computing a computer-generated hologram is shown below:

![Flowchart of the architecture of computer-generated holography](image)

**Step 1:**
In the first step the image is broken down into matrix and then the far field amplitude is calculated by Discrete Fourier Transform. Then the low intensity components are shifted into the centre of the image using another function.

**Step 2:**
In this step a reference wave is created. For creating the reference matrix, the area of the input image is calculated, and a matrix is created which is then filled with zeros. One point in the image is illuminated because we are assuming light coming from a point source. For computational ease the point is considered to be the center of the reference wave.

**Step 3:**
Fourier Transform operation is performed on the reference wave and in the same way like the object wave the low intensity components are shifted to the centre.

**Step 4:**
This step is considered as the superposition of the waves. The far field amplitude of the reference wave and the object wave is added, and we get the final image array. The absolute value of the object image is subtracted by matrix subtraction. The result is the holographic interference pattern.

**Step 5:**
This step is the reconstruction step. In this step Fast Fourier Transform is applied on the matrix that has been obtained from subtracting the object wave from the interference matrix. Then by calculating the absolute value of the matrix the reconstructed image can be obtained. By in this step the image obtained by reconstruction is overlapped by the same image of opposite orientation. This is because two images are created during this process.

2.2 Secret image sharing

With the rapid growth on the internet, it is becoming essential to transfer multiple images at a time without compromising the security. Hence, we are proposing (2n, 2n) (MSIS) scheme consisting of two phases. In our scheme initially, 2n images are encoded into 2n shares using Boolean operation XOR. To enhance the security these shares are further encrypted using the S-Box generated by right side values of K elliptic curve points. In the suggested scheme as described in Figure 3.

![Fig 3: (2n, 2n) Secret Image Sharing Scheme.](image)

dealer has 2n distinct images and encryption is performed by the dealer using secret shared key. Combiner uses the same secret shared key for decryption (to get original images).

2.3 Secret image sharing

Raw data is filtered many times to remove unusually different and abnormal values. The filtered dataset is then obtained with 76755 rows of data. The 76755 rows of data contain one of each of Humidity, Temperature, Light Intensity and Voltage values. Therefore, the set of 4 values defines the uniqueness of each set of data. There can be many values of temperature alone which could be the same. However, a row can only be unique if all the 4 values of Temperature, Humidity, Light Intensity and Voltage together resemble a unique set. Moreover, repetition of rows of data would cause problems when testing an algorithm because same rows of data would mean that an algorithm would have a higher chance of being trained and tested with the same data at the same time. Therefore, it shall give us high yet absurd and unreliable accuracy. This is what the theory of “Problem of Overfitting” explains. Overfitting can occur when an algorithm is trained and tested with the same data leading to a very high accuracy achieved by the algorithm. The figures below show how data would differ after removing identical sets of data. Removal of repetitive rows is necessary for better results. After removal of repetitive data, 69261 data forms. The next process for all the algorithms is the Train - Test split of data. This is same for all them. We never test on the data that we have trained upon. We always try to maintain separate data for training and testing. 80 percent of the entire data is used for training and 20 percent of the entire data for testing. Train and Test split can be obtained automatically by using a built in command in PYTHON. However, train and test data can also be stored in separate.

CSV files which could be used while running the algorithm. After these background work, each algorithm is tested, and different metrics are obtained which points out how successful each algorithm.

2.2 Feature Scaling

When training neural networks, it is considered to be the best practice to scale the data. Otherwise different problems can be encountered such as degraded performance, unpredictable behavior of optimization algorithm and exploding gradient. There are different ways to scale the data, some of them rely on knowing the maximum and minimum values for the distribution, and as the data is not in some predefined range like, e.g. pixel value data, the standardization method is chosen, that relies on computing the mean and variance of the train set.
2.2 Training with Validation, and Testing Sets

For our dataset, we first found the value of ‘k’ that best suited our case. We initially classified with CNN. However, this had no real logic behind it. As a result, we decided to find the optimal value that will give better and correct accuracy. It works by training our model on a range parameter and then finds out the best result based on comparison. In our case, we provided a range of values of k neighbors from 1 to 25 and checked which value of k neighbor provided the most accurate and reliable output. It is seen that best suited our data set. After determining the value, we load the data set. Duplicate rows with same values for all the four features are dropped to avoid training the same data over and over again. It also prevents testing and training with same dataset, avoiding accuracy levels that are unrealistic. All of the data is divided for the different purposes in this segment in which the portion of the data for the training will draw direct conclusions from the validation data. But if the accuracy of the validation set drops below the results of the training data, the network is starting to memorize the data it knows rather than generalizing on the concept.

The third subset is referred to as the testing data. In contrast to the validation set, it’s only used once in the very end, to give a final score. The idea is that by building a model based on the validation results a certain amount of information bleed occurs, where the network will implicitly learn from the validation data. In order to avoid bad results, testing data is normally used as a reference for the performance.

- Training Set: 69% of the data
- Validation Set: 14% of the data
- Test Set: 17% of the data

3. RESULTS

The finest algorithm for intrusion detection and classification on this dataset is CNN. First of all, its accuracy is very high compared to other algorithms. Moreover, the execution time is also very low. As the complexity of the kernel decreases, the algorithm begins to get faster. The other algorithms are more or less same in terms of their execution time. Random Forest has the highest execution time due to the fact that it builds numerous trees before coming to a conclusion or decision. For our experiment we have chosen python as our programming language. Python is a strong programming language and it has several advantages over Matlab because it is free and open source and we can look under the hood how the methods were written unlike Matlab which is closed and proprietary. The python libraries that we needed for our work are:

- numpy
- opencv
- matplotlib

Firstly, we have implemented the process of developing a computer generated hologram and the reconstruction in python. Our algorithm will take the input image and generate a holographic pattern and a reconstructed image from that input image. For breaking down the image into image matrix we have used the OpenCV library. We have used the numpy fft library for the calculation of the far field amplitude of the object and reference wave. After applying fft we need to take the low intensity components to the center of the image for the ease of calculation. We do that by the fftshift() function which is available in numpy. After calculation the far field amplitudes we add the both far field amplitude and we subtract the square of absolute components to the center of the image for the ease of calculation. We do that by the fftshift() function which is available in numpy. After calculating the far field amplitudes we add the both far field amplitude and we subtract the square of absolute value of the object wave, we get a final holographic pattern. In this section, we have considered two parameters while analyzing the Deep Learning algorithm (CNN) on the sensor IoT dataset. They are accuracy in terms of percentage and time in terms of seconds. The experiment is carried using Python Windows environment on 32GB RAM and 2.9 GHz Intel Core i7. After carrying out several experiments using the Convolutional neural network, we have come to present all our findings here. We carried out Convolutional neural network algorithms on our data set. We have done the test procedure with 20 percent data which is equal to 13583 data. The convolutional neural network was created and trained on all 115 features. Default Keras optimization hyper-parameters were used. Training was limited to 100 epochs with additional condition of early stopping, using functionality provided by Keras. This ensures that the model is trained only until the score on validation set is not getting worse – this in turn helps to avoid overfitting the training set. Now we can reconstruct the holographic pattern. We apply Fourier Transform on the final holographic pattern and then shift the low intensity components to the center. Then we find the square of absolute value of our reconstructed image and then we plot the matrix and get the final reconstructed image. In this process we will get two output, the first one is in straight orientation and the other is inverted. The two generated images will overlap with each other in this stage of the process.
The flowchart of our algorithm is shown below:

\[\text{Fig 4: Flowchart of hologram generation and reconstruction.}\]

First an image is taken as an input. Our algorithm will return the holographic interference pattern as an output which can later be reconstructed to form the reconstructed holographic image. For our first experiment let us take the following input image.

\[\text{Fig 5: Input Image}\]

Our algorithm will convert this image into a holographic pattern and then later reconstruct the image. Our algorithm returns the following results,
Fig 6: Hologram and Reconstructed image

From our algorithm we get the two outputs which are shown in the holographic interference pattern which is in left side and in the right side shows the reconstruction of figure one. But from Reconstructed image we see that it is not exactly like our original image. That is because by the theory of holography two images are created during reconstruction. The first one is straight and the second one is inverted. If this matter is not taken care of then parts of image might get overlapped. In our previous example we took a simple image mostly of black parts so the overlapping is not totally visible. Let, we look at another example to figure out what actually can happen when our reconstruction is overlapped. Let us now take an image of a car.

Fig 7: Input image

We take image of a car in figure 8 and run our algorithm and the result is:

Fig 8: Reconstructed Image with overlapping

The reconstructed image of Figure 9 as we addressed earlier this overlapping causes a huge problem while dealing with full sized images. This can be solved manually by editing our input image by cropping it to a quarter now if we do that then it can be helpful but every time we need to manually edit an image which is not a very pleasant thing to do. Then again, the output image will be two with one inverted on the other quarter then again, we have to crop it manually.

Now that we have figured a way to perfectly reconstruct the image from the given image now we will encrypt the image. For encryption let us take our previous example of the car as our sample plain image.

Fig 9: Input Image

Now by the general process of hologram generation we convert the image into grayscale pixel array and perform Fast Fourier Transform on the array. Then to take the DC components to the centre we perform a Fourier shift and get the image in frequency domain.
Figure 10: input image and cipher image

Figure 11 shows a grayscale version of the image that we have used and the right side image is the encrypted image. To reconstruct this image for this encryption we need the encryption key. For decrypting we need to input this cipher image shown in the right side into our decryption algorithm along with our encryption key.

From the encryption key the intensity and the variables that we have used is figured out. The flowchart of the decryption algorithm is shown below:

![Flowchart for decryption of the cipher image using the encryption key](image)

Figure 11. Flowchart for decryption of the cipher image using the encryption key. Below is a demonstration of how our encryption key is broken down into parts.

![Breakdown of the encryption key](image)

Figure 12. Breakdown of the encryption key

Now we have broken down the encryption key into parts. Now we will use these values to calculate our reference wave. With the provided encryption key we reconstruct our cipher image.
The above Figure 15 shows the decryption of the cipher image using the wrong encryption key.

3. Discussion

The cipher images that we have generated appear as black because the intensity values of the Fourier image is too large to be displayed on the screen. We can view the magnitude spectrum of the image by applying logarithmic transformation on our cipher image.

This above Figure 16 is the magnitude spectrum of the cipher image of the car that we have used. Generally when we are reconstructing a holographic interference pattern the final result that we get is basically a hologram which consists of two images with overlapping parts. But we have further developed a scaling technique by which the final reconstructed image is the image itself and not the holographic image. Thus by our algorithm the input image is perfectly reconstructed. Because of our scaling algorithm the encryption and decryption of the image was possible. The most famous image encryption algorithm uses a technique of scrambling the pixels and then the scrambled image is the cipher [8]. This usually has a pattern of pixel scrambling and if the pattern can be figured out then with that pattern the encryption can be broken. In our algorithm we have taken the image from special domain to the frequency domain by applying Discrete Fourier Transform. This can be termed as one level of encryption [9]. To re-transform the image form the frequency domain both the magnitude and phase information has to be preserved. If any of the information is scrambled here then re-transformation is not possible. We have used our second level of encryption here. We took our custom made reference wave where we have scrambled the pixels and took that reference wave to the frequency domain and then added it to the object wave which is in the frequency domain. Now the magnitude and phase data has been scrambled and from here the image cannot be re-transformed. From here we have generated the holographic pattern and as we said earlier because our scaling algorithm allows us to get the image by
reconstructing the interference pattern. Now what makes our encryption stronger than the other image encryption techniques is that while in the frequency domain it is not possible to find a pattern because the coordinates are so large and has no specific pattern. So our cipher image cannot be re-transformed, it has to go through the reconstruction process for getting back the original image.

5. Conclusion

The above techniques are currently applicable in the context of 2D images. But the algorithm is general enough to take a 3D object as an input and encrypt the 3D object using the same technique, with minor tweaks in the code. The techniques described in this paper can also be used to overcome some basic problems which is faces during the reconstruction of image by Computer Generated Holography (CGH) technique such as the image overlapping problem that occurs during the reconstruction of the original image from a hologram. These techniques have a wide range of utilities in digital forensics and security domain. Image encryption or hiding is an important practical problem and the CGH technique described in this paper provide a computationally fast and inexpensive way to generate a strong encryption of an image. The encrypted images that is generated by this technique cannot be decrypted without the encryption key because by the application of Discrete Fourier Transform the image matrix gets one level of encryption and then we have applied a second level of encryption to this image. This technique can be used in a wide range of security applications in the future. Our holographic interference pattern generation algorithm can also be used to generated holograms from any given image and with the use of hologram generation instruments and lasers, these holograms can be printed in any holographic plate. Our algorithm is applicable only for grayscale images for now but in future it can be developed so that coloured images can be transformed into holographic interference pattern and can be encrypted using our technique and with this technique we achieved accuracy of 94.20%.

References

[1] Alvarez, G.; Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. Int. J. Bifurc. Chaos 2006, 16, 2129–2151.
[2] Chai, X.; Chen, Y.; Broyde, L. A novel chaos-based image encryption algorithm using DNA sequence operations. Opt. Lasers Eng. 2017, 88, 197–213.
[3] Hsiao, H.-I.; Lee, J. Color image encryption using chaotic nonlinear adaptive filter. Signal Process. 2015, 117, 281–309.
[4] Mirzaei, O.; Yaghoobi, M.; Irani, H. A new image encryption method: Parallel sub-image encryption with hyper chaos. Nonlinear Dyn. 2012, 67, 557–566.
[5] Wu, Y.; Zhou, Y.; Agaian, S.; Noonan, J.P. A symmetric image cipher using wave perturbations. Signal Process. 2014, 102, 122–131.
[6] Patidar, V.; Pareek, N.K.; Purohit, G.; Sud, K.K. Modified substitution-diffusion image cipher using chaotic standard and logistic maps. Commun. Nonlinear Sci. Numer. Simul. 2010, 15, 2755–2765. [CrossRef]
[7] Zhu, C. A novel image encryption scheme based on improved hyperchaotic sequences. Opt. Commun. 2012, 285, 29–37. [CrossRef]
[8] Zhang, Y.; Xiao, D.; Shu, Y.; Li, J. A novel image encryption scheme based on a linear hyperbolic chaotic system of partial differential equations. Signal Process. Image Commun. 2013, 28, 292–300.
[9] Zhang, W.; Wong, K.-W.; Yu, H.; Zhu, Z.-L. A symmetric color image encryption algorithm using the intrinsic features of bit distributions. Commun. Nonlinear Sci. Numer. Simul. 2013, 18, 584–600.
[10] Zhang, Y.; Xiao, D. An image encryption scheme based on rotation matrix bit-level permutation and block diffusion. Commun. Nonlinear Sci. Numer. Simul. 2014, 19, 74–82.