IMAGE ENCRYPTION USING HOPFIELD NEURAL NETWORK

ENM 531, SPRING '23 - DATA DRIVEN MODELLING

DHRUV GUPTA [DGUPTA99@SEAS.UPENN.EDU], RAJNISH GUPTA [RAJNISHG@SEAS.UPENN.EDU],

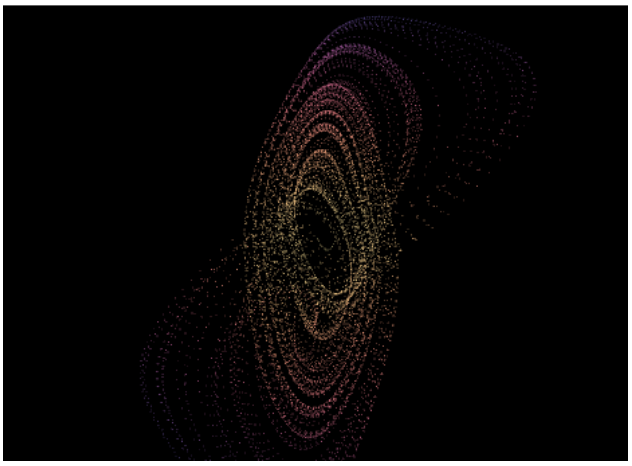

FIGURE 1. Chaotic Hopfield Network Sequence in 3-D space

1. INTRODUCTION

Encryption of data is crucial in protecting sensitive information from unauthorized access or theft. With the growing reliance on digital systems in various industries and markets, encryption technology has become an indispensable tool for safeguarding confidential data, ensuring privacy, and maintaining trust among customers and stakeholders. The best way to store any sensitive information is to store it on a device completely disconnected from the rest of the world, but that isn't an ideal case, and more importantly, security is needed while sharing that data/information. In this project we would mainly focus on images, and how we can encrypt them over transmissions. While there are various encryption techniques, not all of them provide the same level of security. We need to insert some level of randomness that can make the encryption immune to brute-force attacks; and one way of doing this is by introducing chaos into our system (pseudo-randomness). In this implementation of an optical encryption system, we use two forms of dynamical systems that introduce chaos; the Hopfield Neural Network and the Single-Neuronal Dyanmic System.

2. CHAOTIC SYSTEM

Chaos-based cryptography is an extremely useful technique for encryption of images. These rely on dynamical systems that are highly sensitive to initial conditions and can thus be resilient to brute-force attacks on encrypted data.

Chaotic maps generate pseudo-random sequences - in that if the initial conditions and configuration of the system is unknown, it can be extremely difficult to reverse-engineer the system from the output sequence and also predict any numbers given existing data. This property of chaos can be used in encryption to generate pseudo-random sequences that can be used to scramble data. The encryption scheme we studied, makes use of two such chaotic systems detailed below.

2.1. **Hopfield Neural Network.**

A Hopfield neural network is a single-layer recurrent neural network that has the ability to recognize or remember patterns. A schematic is shown in figure 1. Typically, a Hopfield network would not have weights associated with the synapses that feed back into the neuron (w_{ii}). However, to show chaotic behaviour we

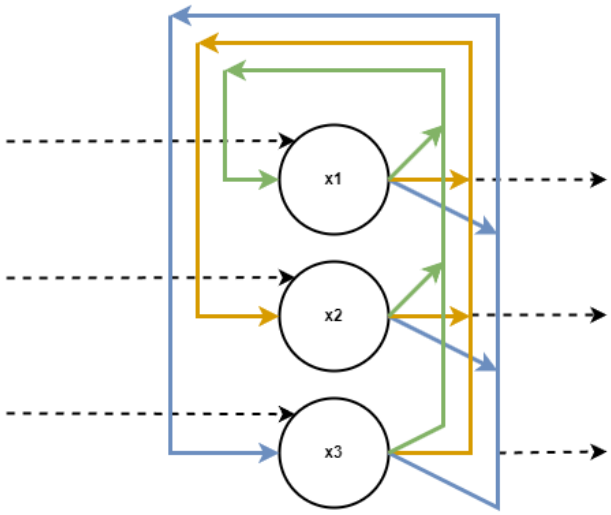

FIGURE 2. Hopfield Neural Network Architecture

make use of a modified form of the network - a three-dimensional cellular network with an asymmetric synaptic weight matrix. The network follows simple update rules

$$\dot{x} = -x + W \cdot \phi(x) \tag{1}$$

where x is the neuron state vector (3 dimensional in our case), and ϕ is the activation function

$$\phi(\mathbf{x}) = \tanh \mathbf{x} = [\tanh \mathbf{x_1}, \tanh \mathbf{x_2}, \tanh \mathbf{x_3}] \tag{2}$$

W is the synaptic weight matrix. The Hopfield network chosen for this encryption has a synaptic weight matrix, obtained through specific training, that has verifiable chaotic behaviour.

$$W = \begin{pmatrix} 2 & -1.58 & -0.27 \\ 1.87 & 1.71 & 1.04 \\ -6.92 & -0.58 & 1.1 \end{pmatrix} \tag{3}$$

Figure 2 displays the network's dynamics for initial conditions that display chaotic behaviour. As is apparent, the

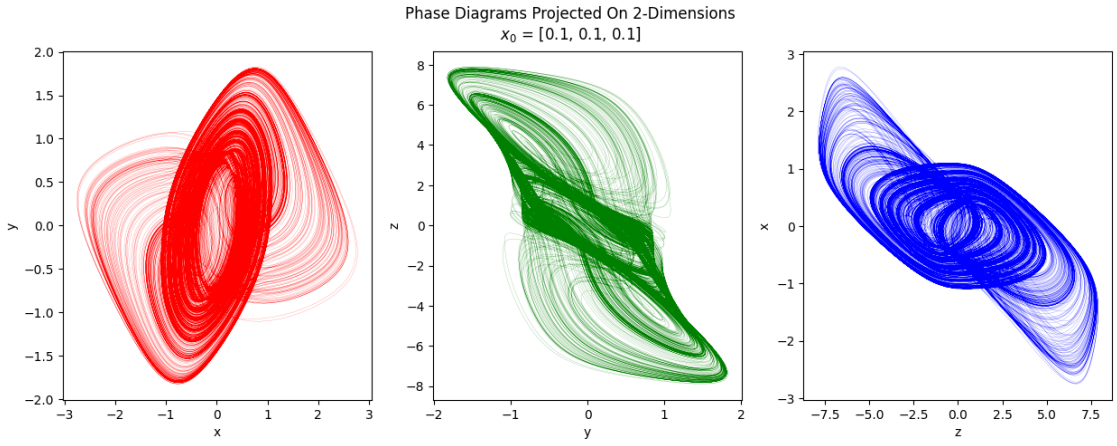

FIGURE 3. HNN Dynamics showing a trajectory of 50,000 points

trajectory is periodic and travels around the attractors in the phase space. Certain configurations of the network will thus result in chaotic behaviour if the initial conditions or weights are perturbed. Further confirmation of chaotic behaviour of these networks would require analysing their Lyapunov exponents, which have been studied extensively.

2.2. **Single Neuronal Dynamical System.** A single-neuronal system is a derivative of the Hopfield network with a single neuron and sigmoid activation.

$$v(t) = \frac{1}{1 + exp\gamma u(t)} \tag{4}$$

$$v'(t) = v(t) \cdot 2^n - \lfloor v(t) \cdot 2^n \rfloor \tag{5}$$

$$u(t+1) = ku(t) + zv'(t) + h \tag{6}$$

u and v are the input and output signals of the network respectively, while $u(0), \gamma, n, k, z, h$ are hyperparameters. The particular configuration used is chaotic for certain intervals in the parameter space and that region is what ensures data transmission security in our encryption scheme.

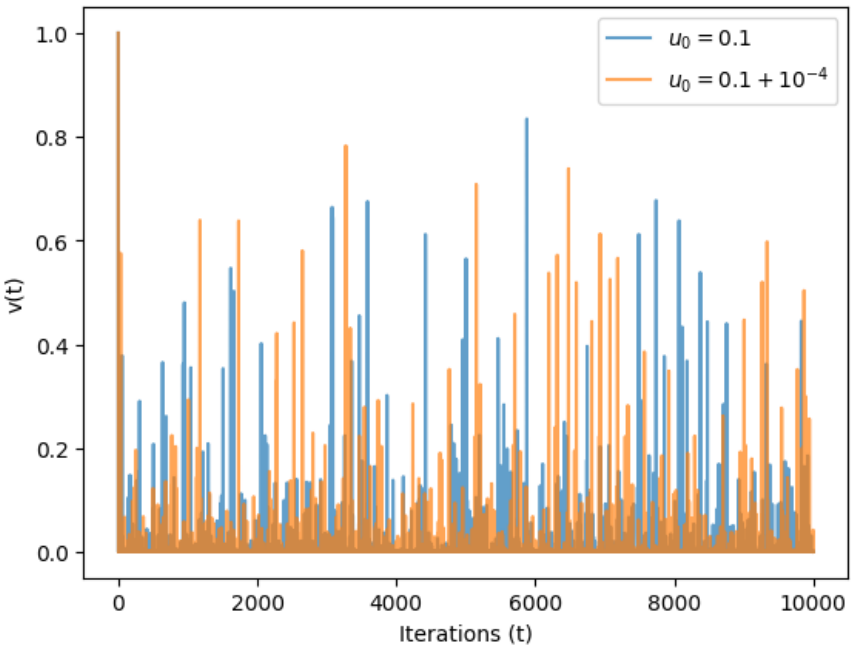

FIGURE 4. Sensitivity of the Single Neuronal Dynamical System to Initial Conditions

3. OPTICAL SYSTEM

In our encryption method, we have used various mathematical functions, which combined together to work as an optical encryption system - for which the physical analogue is lens system that scrambles the signals differently based on their phase-shifts. An important property which we can exploit for an optical system is to decompose the image into subsignals that can be processed parallely.

3.1. **Wavelet Packet Transform.** Wavelet transformation has multiple uses - in particular in our case, it helps us decompose the signal into sub-signals, of different sizes, and also compress it, making our implementation more effective and efficient. We would use **Daubechies Wavelet transformation**, and implement it using an order of 2. It converts the image into a composition of a mother-signal, and a set of residual signal, which can be used to reconstruct the image using inverse wavelet transformation. The decomposed image can also be easily compressed. The residuals are the differences between the adjacent pixels on each level, and the mother wavelet is the re-scaled image to a smaller size, which has the average of a set of pixels.

3.2. **4f System/Fourier Transform.** After we obtain the sub-signals, and scramble them with a chaotic system and through Arnold scrambling, we then use two different functions involving various optical transformations. We separate signals with more information in one layer, and signals with very small information(small variance), through a different

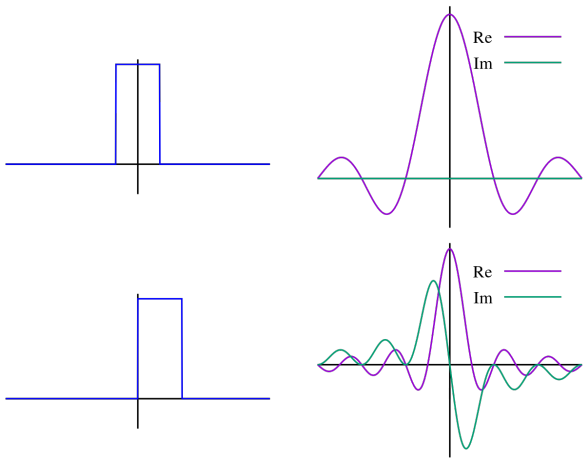

FIGURE 5. Fourier transformation

layer, and perform two different sets of transformation techniques. The main sub-signal (high variance part of the signal) is processed by a 4f system Φ_{4f}:

$$\Phi_4 f = FT^{-1}[FT[P_t * C_1 t] * C_2 t]$$

using Fourier, and inverse Fourier transformation, where P_t are the sub-signals obtained from the wavelets, and C_1, and C_2 are the complex chaotic system we obtained from X_2 and X_3. It would also result in some imaginary values, which are discarded when we present the image after inverse wavelet transformation, although they are useful in obtaining back the the original image.

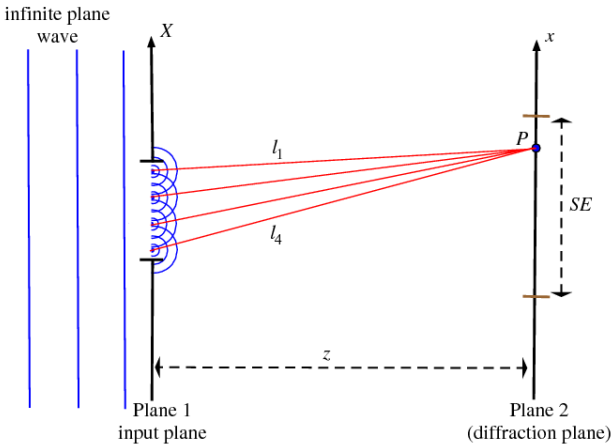

FIGURE 6. Fresnel transformation

3.3. **Fresnel Transform.** The remaining part of the signal is transformed using a set of Fresnel transformations. Similar to the function given above we would apply fresnel transformation, but in this case, instead of an inverse Fourier, we would complement it with another fresnel transformation, and obtain the transformation function $\Phi_{fresnel}$ as:

$$\Phi_{fresnel} = FrT_{\rho,d2}[FrT_{\rho,d2}[P_t * C_1 t] * C_2 t$$

In this, the $d1$, and $d2$ are the distance between the input and diffraction plane, and ρ is the wavelength of the light used in fresnel transformation. After we obtain the signal from the transformation, we combine them to get an inverse wavelet transformation, to get an encrypted image. The pipeline describes this in extreme detail.

4. Pipeline and Methodology

We start with an image ($M \times N$) that must be encrypted. The encryption scheme interleaves the chaotic system with the optical encryption mechanism. We use the following steps.

4.1. **Wavelet Transform.** The image is decomposed into sub-signals which projects the image into a basis of 'wavelets'. In the 2D case, for an 'm' order wavelet transformation, we obtain 'T' sub-signals, where $T = 3m + 1$. For our purposes we used the second-order decomposition from the PyWavelets Python package. This gives us 7 sub-signals, as seen in the figure.

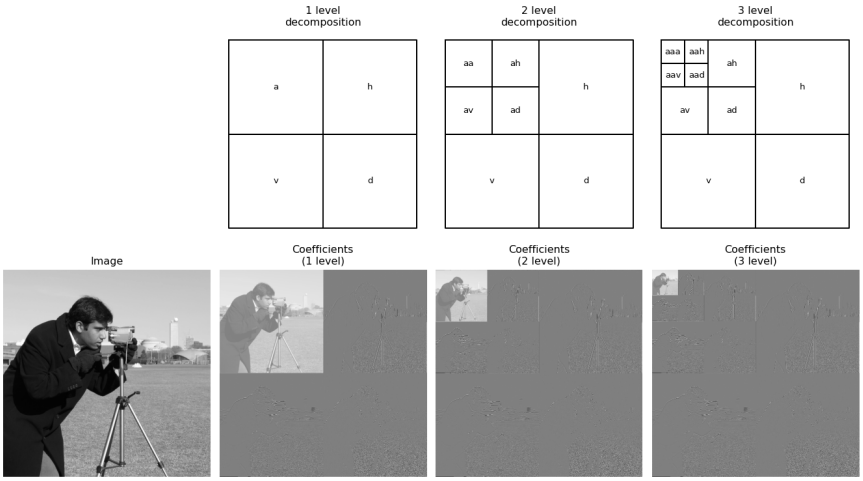

FIGURE 7. Wavelet Transformation For Multiple Orders - PyWavelets

At the same time, we now obtain an $M \times N$ long sequence from the Hopfield Neural Network, using initial values of x_1, x_2, x_3. These are part of the private keys of the encryption.

4.2. **Sequence Shuffling.** We assign a state variable to each point in the sequence

$$S = mod(\lfloor \sqrt{x_1^2 + x_2^2 + x_3^2} \rfloor, 3)$$

And we assign an order of the sequence based on the value of the state variable. For S=0, x_1, x_2, x_3 is added; for S=1, x_2, x_3, x_1 is added; and for S=2, x_3, x_1, x_2 is added to the new sequence. Each of the new sequences obtained X_1, X_2, X_3 are used henceforth.

4.3. **Sub-Signal Shuffling.** We now use X_1 as an index to scramble the sub-signals. X_1 is grouped into T sub-sequences $L = [L_1, L_2, ..., L_T]$, and the sub-signals are flattened to $P = [P_1, P_2, ..., P_T]$.

For each sub-sequence L_t, its elements are sorted in ascending order, and the corresponding flattened sub-signal P_t is re-ordered in the same way. These are then converted back to 2-D arrays as modified sub-signals.

4.4. **Sigma-Sorting.** The modified sub-signals will undergo one of two optical transformations. This is decided by obtaining their standard deviations $\sigma = [\sigma_1, ..., \sigma_T]$; all sub-signals with $\sigma_t \geq \bar{\sigma}$ will go through the 4f transform layer, and the others will undergo Fresnel transform layer, where $\bar{\sigma} = \frac{1}{T} \sum \sigma$

4.5. **Arnold Scrambling of Chaotic Matrices.** The remaining chaotic sequences, X_2, X_3, are converted to T sub-sequences, as was done in step 1 with X_1, and are converted to 2-D matrices with the same shape as the corresponding sub-signals. For example, for an image size of 512*512, the first sub-signals has dimensions 128*128. The first chaotic matrix will have the same dimensions. Following this, we scramble the elements of the chaotic matrices using Arnold Scrambling

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = mod \left[\begin{pmatrix} 1 & \alpha \\ \beta & \alpha\beta + 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, N \right] \tag{7}$$

where (x,y) are the original coordinates in the chaotic matrix, and (x',y') are the scrambled coordinates. α, β are the parameters of the scrambling algorithm, and N is the size of the matrix.

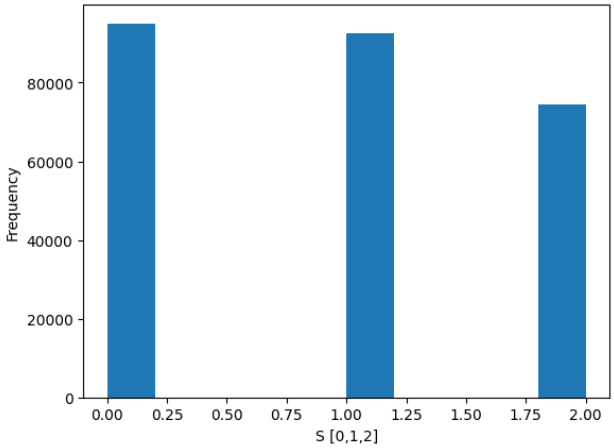

FIGURE 8. Distribution of the State Variables For The Hopfield Chaotic Sequence (For a 512*512 image)

4.6. **Random-Phase Creation.** The chaotic matrices are normalized to obtain $g_t(x,y), r_t(x,y)$ from X_2, X_3 respectively, and used to construct the random phase shifts for the sub-signals.

$$C_{1t} = \exp[i2\pi g_t(x,y)]$$
$$C_{2t} = \exp[i2\pi r_t(x,y)]$$

4.7. **Optical Transforms.** The 4f transform is performed on the first set of sub-signals obtained from sigma sorting and the Fresnel transform on the second set.

$$\Phi_{4f} = FT^{-1}[FT[P_t(x,y) * C_{1t}(x,y)] * C_{2t}(x,y)]$$
$$\Phi_{Fresnel} = FrT_{\rho,d1}[FrT_{\rho,d2}[P_t(x,y) * C_{1t}(x,y)] * C_{2t}(x,y)]$$

Using the chaotic phases C_{1t}, C_{2t} scrambles the coordinates through the optical transformations. The fresnel transformation uses the wavelength ρ, and diffraction distance d_1, d_2 as parameters.

4.8. **Inverse Transform.** The now-confused sub-signals are recombined using the inverse wavelet transform to form the image E, which is normalized and scrambled using the single-neuronal dynamical system to obtain the encrypted image.

4.9. **Deciphering.** The decryption will require knowledge of the initial conditions and parameters of the neuronal dynamic system and Hopfield network - $u_0, \gamma, n, k, z, h, s, x_1(0), x_2(0), x_3(0)$.
Thus, given the initial state and parameters of the system of HNN and SNDS, we can re-generate the chaotic system used to encrypt the image and perform the inverse transformations on the cipher data. It is fairly direct for a Fourier transformation since we only have to use inverse Fourier and unscramble the generated values with the chaotic matrices. While working with the $\Phi_{fresnel}$, we would have to do inverse fresnel transformation, which is given by:

$$x = \frac{\lambda}{2\pi}\int_0^z du\cos(\frac{\pi u^2}{\lambda d})$$
$$y = \frac{\lambda}{2\pi}\int_0^z du\sin(\frac{\pi u^2}{\lambda d})$$

After we are done with this, we just need to unscramble using the chaotic matrices, rearrange and reshape to obtain the original wavelets, and then perform a wavelet transformation to obtain the original image.

5. RESULTS

Here we have taken a sample image, which you might be familiar with as it is commonly used in many computer vision papers, we have successfully transformed it into an encrypted image, as seen below: .

FIGURE 9.
Orignal Image

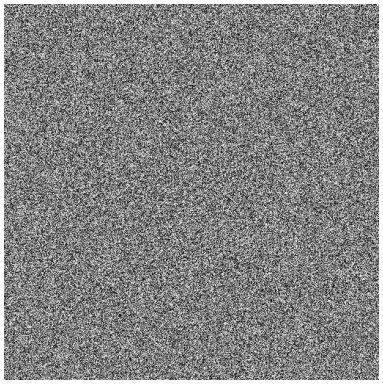

FIGURE 10.
Encrypted image

FIGURE 11.
Decrypted Image

We are successfully able to generate the encryption and decryption as shown here. We can see that the decyphered image is greyscaled, that is because we applied the encryption after converting the orignal image to greyscale, so we only have to work with 1 channel, but we can apply the same, system on all the three RGB channels, and we will be able to obtain an encrypted coloured-image.

6. DISCUSSION

While we were not able to test the efficacy of the algorithm against brute-force or general decryption attacks, references cited that have used this technique proves that it holds up to modern encryption standards. In practice, the private keys of the encryption (parameters of the neuronal system and Hopfield network) will be transmitted through RSA or other existing cryptographic techniques. There is a lot of room for improvement in the choice of dynamic systems - such as using a logistic map instead of the single-neuronal system, or a different activation function for the Hopfield network.

6.1. **Alternative Methods - Feed Forward Neural Networks.**

An alternate cryptographic technique that makes use of Neural networks uses the synchronization of tree Parity Machines. To do so, two feed-forward neural networks are trained simultaneously on training data (encrypted and decrypted), while optimizing for the accuracy of the desired cypher texts. Once synchronized, these networks can be used to exchange encryption keys. Brute-forcing to decipher the data can be extremely difficult without knowledge of the architecture and weights of the network.

7. ACKNOWLEDGEMENTS

Thank you Dr. Paris and Shyam Sankaran, the instructor and TA for ENM 531, respectively. It has been truly enlightening exploring a completely new application of neural networks through the project, and also get a deeper look into data modelling and machine learning.

REFERENCES

- *Xitong Xu and Shengbo Chen,* An Optical Image Encryption Method Using Hopfield Neural Network
- *Xiao-Song Yang, Quan Yuan,* Chaos and transient chaos in simple Hopfield neural networks
- *Xiao-Song Yang and Qingdu Li*, HORSESHOE CHAOS IN CELLULAR NEURAL NETWORKS
- *Xitong Xu and Shengbo Chen*, Single Neuronal Dynamical System in Self-Feedbacked Hopfield Networks and Its Application in Image Encryption
- *Chen Liang, Qun Zhang, Jianfeng Ma and Kaiming Li,* Research on neural network chaotic encryption algorithm in wireless network security communication
- *Muhammed J. Al-Muhammed and Raed Abu Zitar,* Encryption technique based on chaotic neural network space shift and color-theory-induced distortion
- *Ying Mao,* Algorithm of Encrypting Digital Image Using Chaos Neural Network
- *Shuying Wang, Ling Hong, Jun Jiang,* An image encryption scheme using a chaotic neural network and a network with multistable hyperchaos

- *Behrouz Zolfaghari and Takeshi Koshiba,* The Dichotomy of Neural Networks and Cryptography: War and Peace