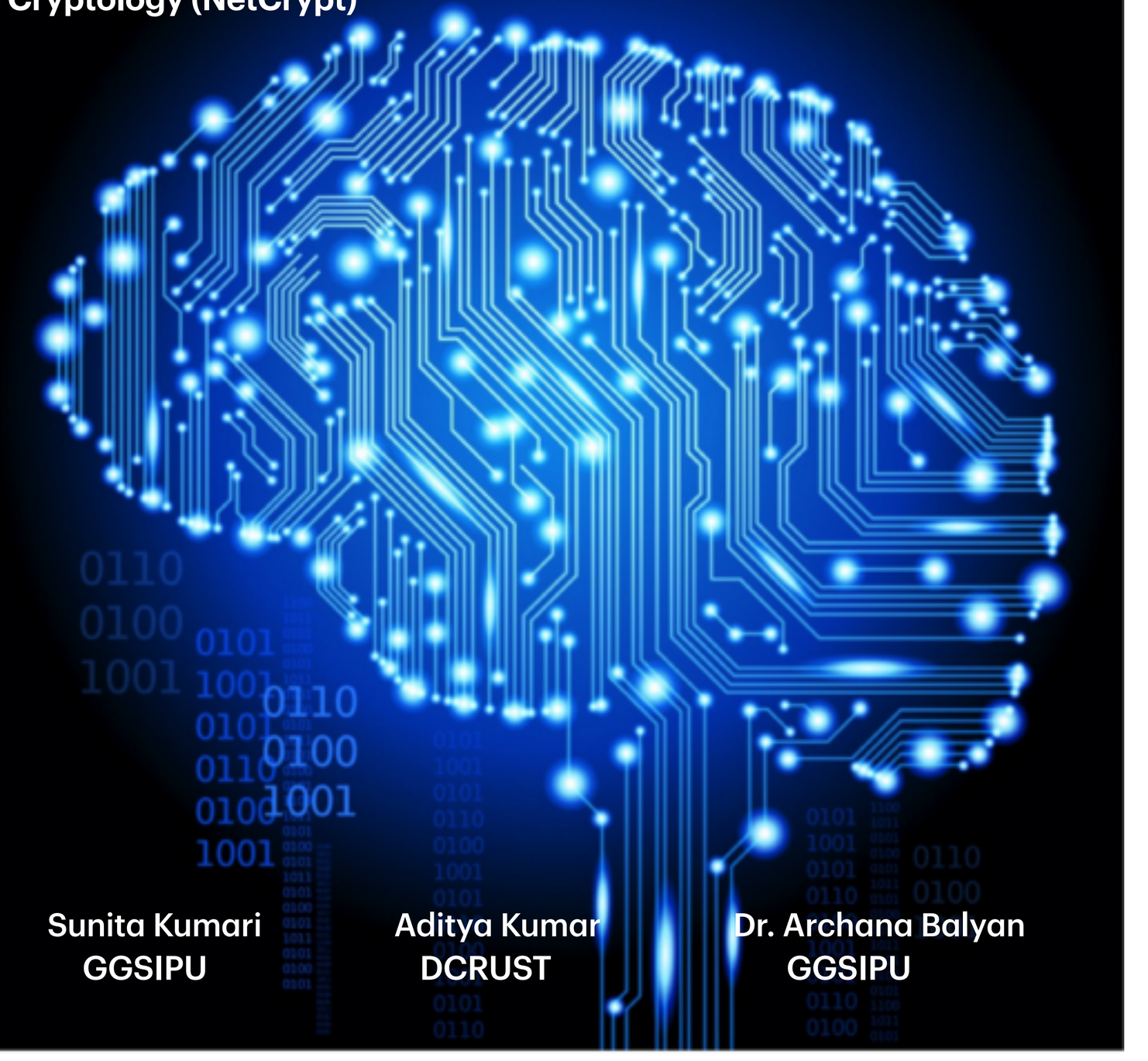
The 2nd International Conference on Networks and Cryptology (NetCrypt)
Jawaharlal Nehru University, New Delhi, India
4-6 December, 2020



GENigma(MayhemNet): Canalizing Stochasticity of Neural Nets into an Insurmountable Encryption Machine

Peeyush Kumar
GGSIPU

Ayushe Gangal*
GGSIPU

Sunita Kumari
GGSIPU

Aditya Kumar
DCRUST

Dr. Archana Balyan
GGSIPU

CONTENTS

INTRODUCTION

- With the exponential rise in the usage and dependency on data in today's world, this data's security and confidentiality is imperative.
- Image cryptography and steganography are two such methods by which data can be secured.
- The former uses mathematical operations to conceal the message and hides its true form, while the latter uses an image to physically camouflage it.
- The proposed encryption machine called GENigma, which uses image steganography and neural cryptography as its base element.
- The initial inspiration for GENigma comes from "Enigma", an encryption machine used in the early-to-mid century by the German forces during world war II.

EARLIER WORKS

- R Ramamurthy et al. [1] proposed a new approach and used echo state networks as both encryptors and decoders. The results of image encryption and decryption were shown, with differences of 33.22% and 37.78% respectively, between the two sets of images for plaintext sensitivity.
- E Volna et al. [2] used ANN for cryptography. Multilayer networks adapted using back propagation were used for encryption and decryption of the plain text data. The network consisted of 6 nodes each in the input and the output layers. Though, the system was found to be noise tolerant, but could only be used for plain text data.
- T Godhavari et al. [3] performed synchronization by mutual learning of two separate neural networks for successful encryption and decryption. The DES algorithm was used for encryption and decryption, and was simulated using VHDL.
- J Hayes et al. [4] used unsupervised learning algorithm for producing steganographic images. Adversarial training method is used for both unsupervised and supervised method. The unsupervised learning method was used to generate steganographic images, while supervised learning method was used by the steganalyzer for the purpose of detecting the presence of any secret message in the image.

PROBLEM STATEMENT

Conventional data encryption approaches using neural-cryptography and steganography:

- Do not produce 100% data retrieval rates
- Require large amount of data to train
- Have long training time
- Offer least to none customizability

CONTRIBUTION

- The proposed encryption system called GENigma consists of a neural encryption network named MayhemNet and an intelligently trained generative network named Generative Empirical Network or GEN.
- GENigma consists of three neural networks, namely:
  - ➡ The MayhemNet encoder: encrypts the secret message and does not require any training, thus reducing the overall training time of the GENigma.
  - ➡ Generative Empirical Network (GEN): generates the artistic cover image for performing image steganography.
  - ➡ The MayhemNet decoder: decrypts the encrypted message.

PROPOSED METHODOLOGY

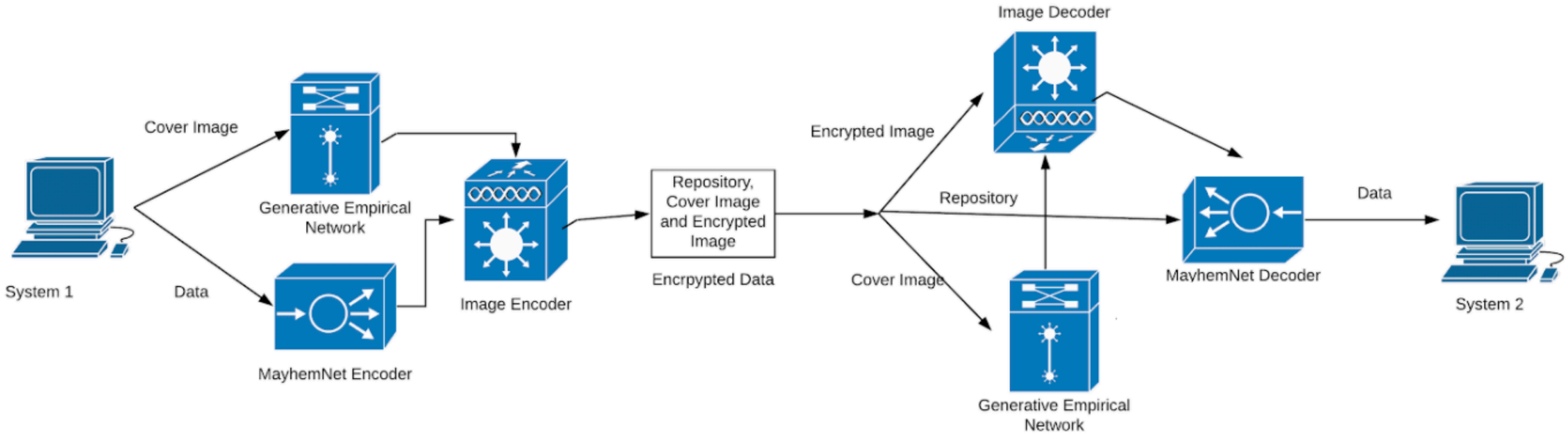


Fig 1: Flow Diagram of the proposed encryption system.

PROPOSED METHODOLOGY

MayhemNet has 3 layers:

- Data to Numbers Mapping Layer represented as:

$$E^{M}_{P \rightarrow Numerical}(P) = P_{Numercial} \tag{1}$$

- Dense Encryption Network Layer represented as:

$$E^{W,F}_{Dense}(P_{Numerical}) = P_{Dense} \tag{2}$$

- Sampling Layer represented as:

$$E^{W_s,F}_{Sampling}(P_{Dense}) = P_{Sampling} \tag{3}$$

- The whole MayhemNet can be represented as :

$$E^{W_s,F}_{Sampling}(E^{W,F}_{Dense}(E^{M}_{P \rightarrow Numerical}(P))) = Encrypted(P) \tag{4}$$

CONTINUED

Generative Empirically Network (GEN) can be Mathematically represented as follows :

- Let $G_b^a(I, \Theta)$ be the generator that trains from iteration number *a* to iteration number *b* using training set *[I, I]* and tries to map *I* to *I* by using, as well as simultaneously learning and updating parameters Θ. At iteration *b*, Generator G will perform the test using original image I , which is the initial input from the users to GEN, and the trained parameters Θ_b^T as *G(* I , Θ_b^T) and test result I_b^T is outputted along with trained parametersΘ_b^T. Such a generator *G* can be represented as:

$$G_a^b(I, \Theta) = I_b^T, \Theta_b^T \tag{5}$$

- For the whole training period of 0 to 600 iterations, the equation for the whole training process using equation (5), for image I provided by the user and for initialized parameters Θ , can be given as:

$$G_{500}^{600}(G_{400}^{500}(G_{300}^{400}(G_{200}^{300}(G_{100}^{200}(G_0^{100}(G_0^0(I, \Theta)))))))) = I_{600}^T, \Theta_{600}^T \tag{6}$$

The rest of the generated images in equation (6) will be as follows: $I_0^T, I_{100}^T, I_{200}^T, I_{300}^T, I_{400}^T, I_{500}^T, I_{600}^T$

CONTINUED

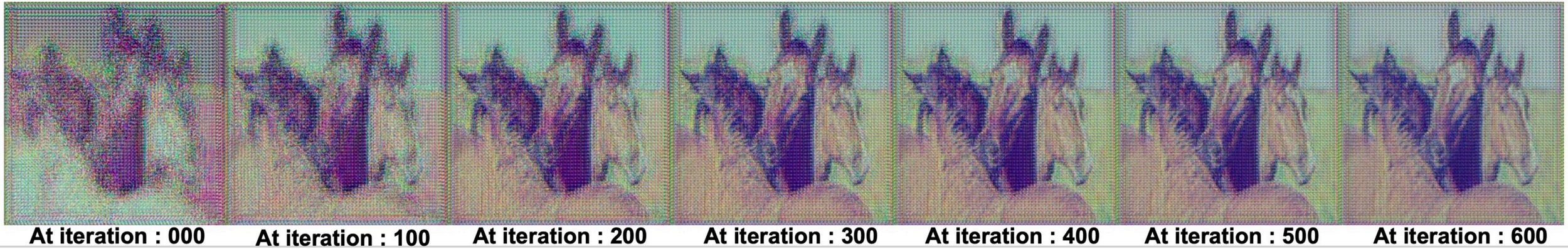


Fig 2: Shows the association between the image generated and the number of iterations

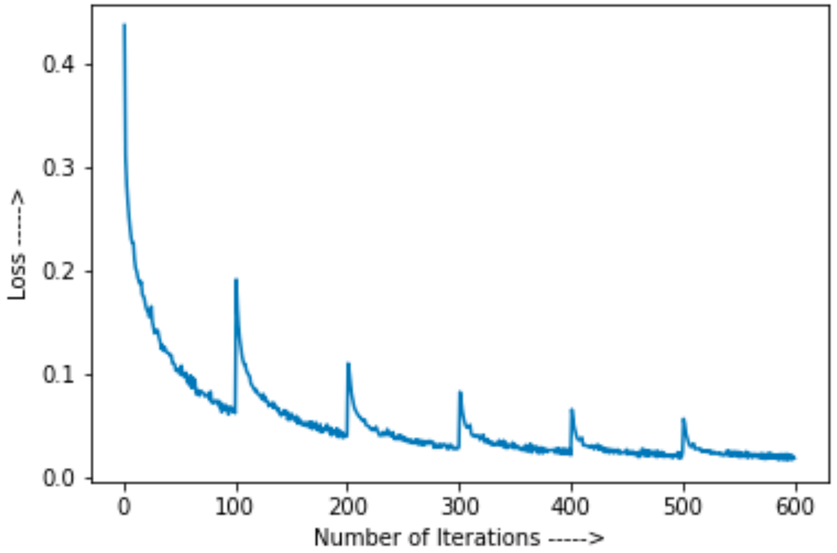
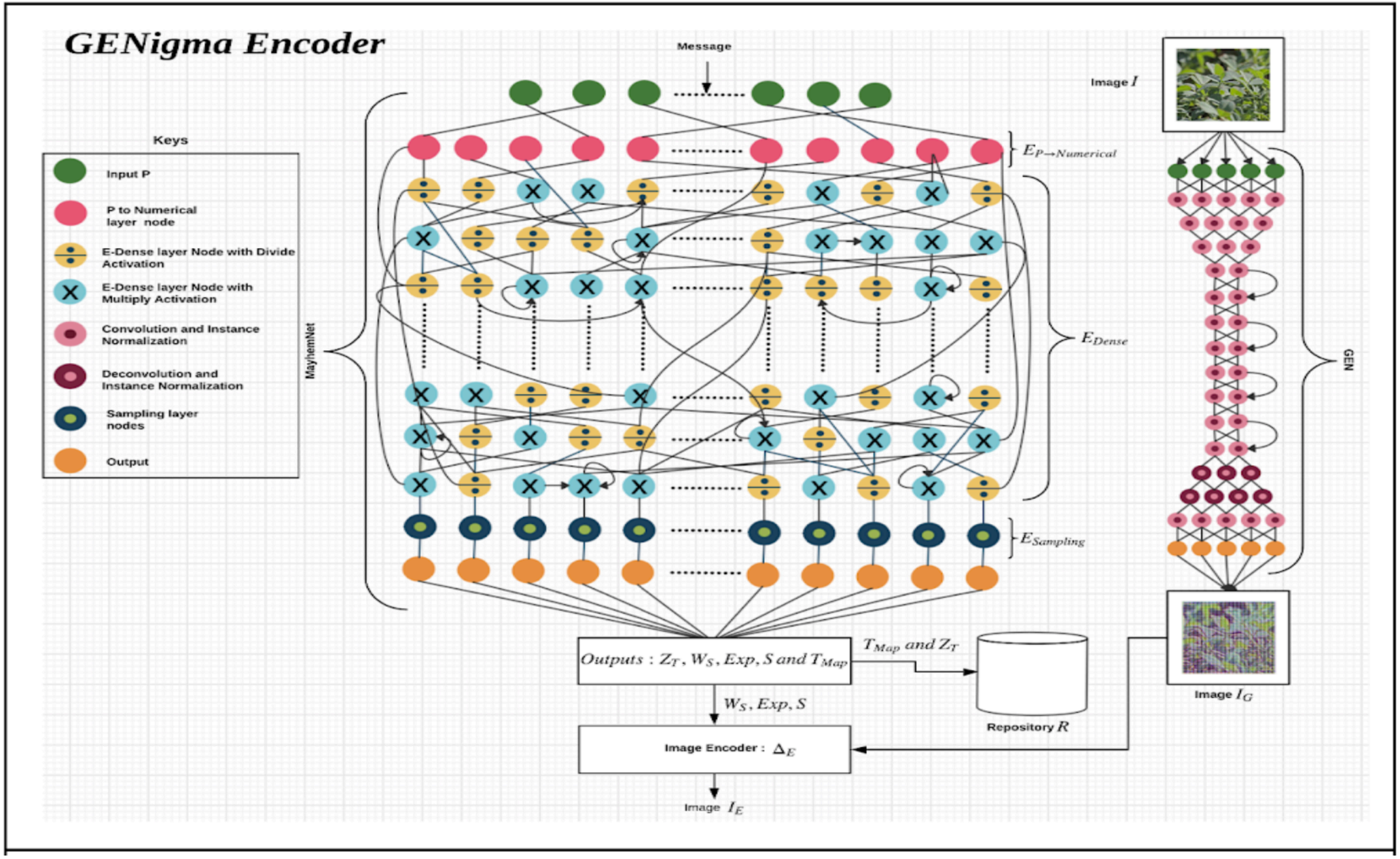
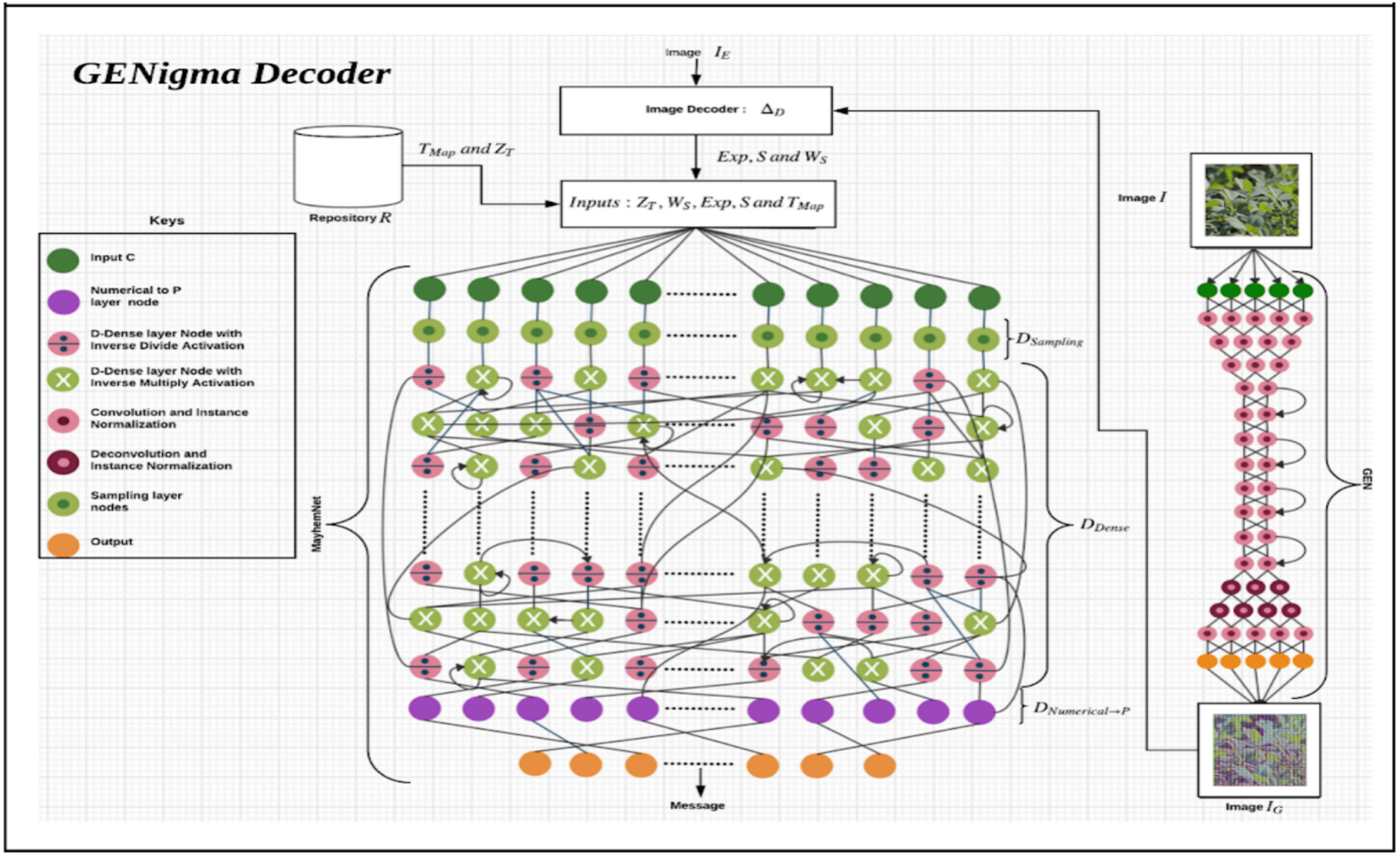


Fig 3 : The graph shows the trade off between loss and iteration number. Each bulge in loss value is due to the change of training set at each checkpoint

ARCHITECTURE



CONTINUED


GENigma Decoder
Image I_E
Image Decoder : Δ_D
T_{Map} and Z_T
Exp, S and W_S
Repository R
Inputs : Z_T, W_S, Exp, S and T_{Map}
Image I
Keys
Input C
Numerical to P layer node
D-Dense layer Node with Inverse Divide Activation
D-Dense layer Node with Inverse Multiply Activation
Convolution and Instance Normalization
Deconvolution and Instance Normalization
Sampling layer nodes
Output
MayhemNet
$D_{Sampling}$
D_{Dense}
$D_{Numerical \to P}$
GEN
Message
Image I_G

RESULTS AND DISCUSSIONS

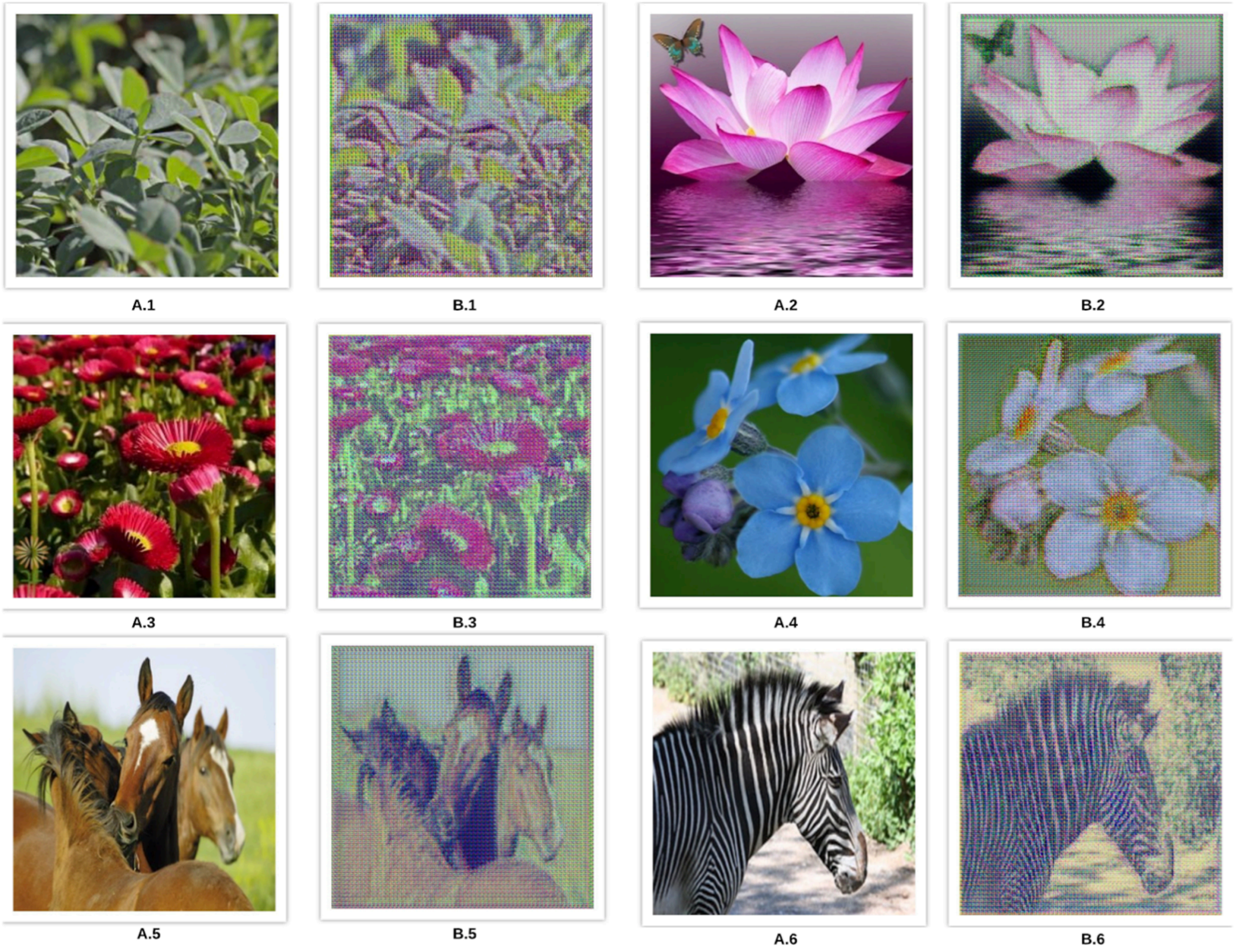


Fig 4: Shows Original images (A's) and their respective 'generated + encoded' images (B's). The text message "*Co-ordinates=40.714,-74.006*" was encoded using MayhemNet architecture given in fig 1 with maximum number jumps allowed = 100, and the output was then passed into the image encoder, proposed in the paper, along with images generated by GEN

CONTINUED

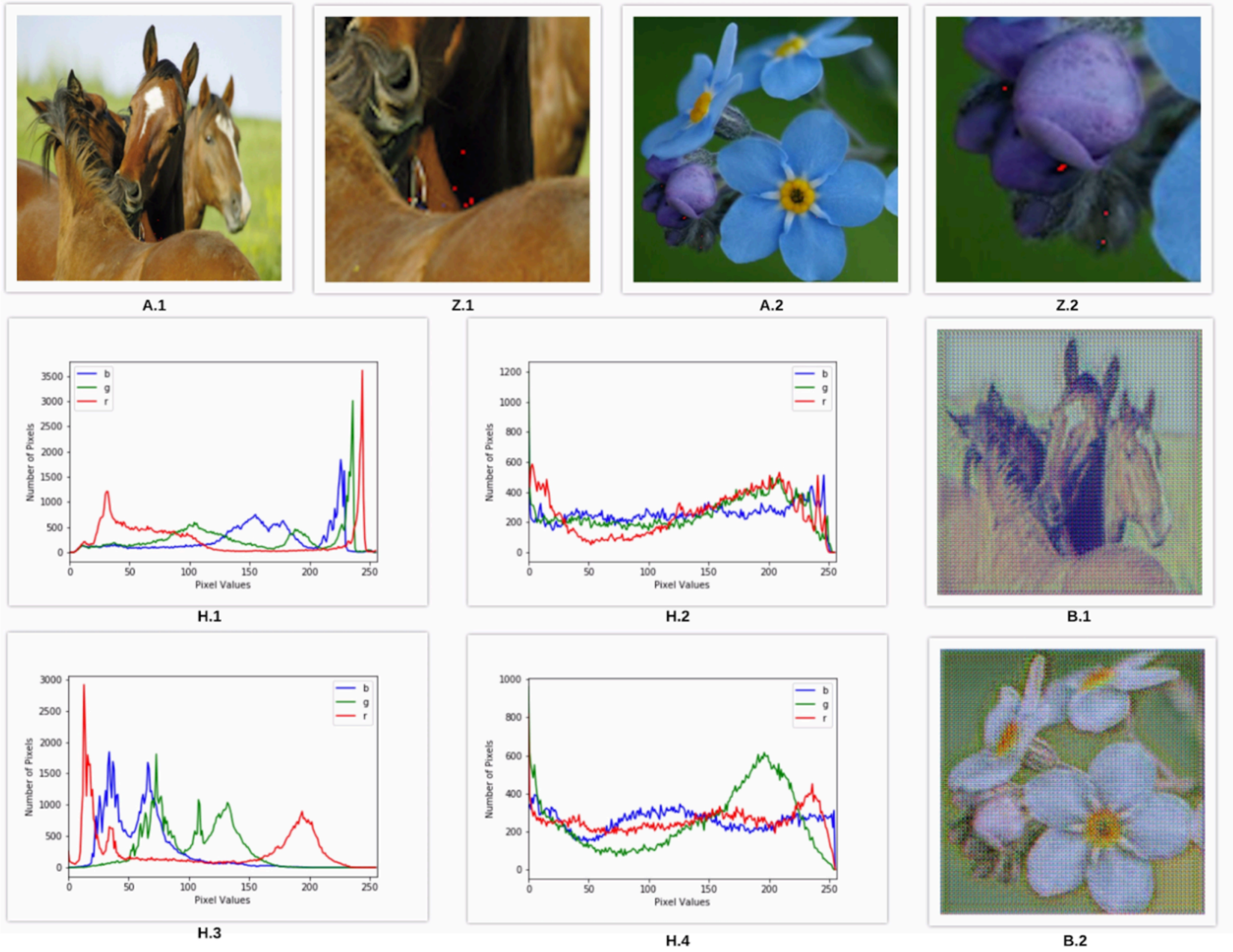


Fig 5: Shows histograms of images A.1, A.2, B.1, B.2 in images H.1, H.3, H.2, H.4 respectively. Images Z.1 and Z.2 shows the zoomed-in region of images A.1 and A.2, where the defects have appeared due to encoding

CONCLUSION

- The proposed system has 100% data retrieval rates at the decoding end.
- The proposed system offers high customizability, as the user can plug-in their own custom made image encoder instead of using the one provided.
- The GEN only needs one image for training, unlike the approaches seen so far, which required a large dataset.
- Every part of GENigma machine is highly complex, highly customizable and utterly implausible to breach.

REFERENCES

1. Ramamurthy R, Bauckhage C, Buza K, Wrobel S (2017) Using echo state networks for cryptography. In: Proceedings of *International Conference on Artificial Neural Networks*, pp. 663-671. Springer, Cham.
2. Volna E, Kotyrba M, Kocian V, Janosek M (2012) Cryptography Based on Neural Network. In: proceedings of ECMS. 386-391.
3. Godhavari T, Alamelu NR, Soundararajan R (2005) Cryptography using neural network. In: 2005 Annual IEEE India Conference-Indicon, pp. 258-261.
4. Hayes J, Danezis G (2017) Generating steganographic images via adversarial training. In Adv. Neural Inf. Processing Sys., pp. 1954-1963.

THANK YOU