ISSN 2278-3091
Volume 10, No.3, May - June 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse451032021.pdf
https://doi.org/10.30534/ijatcse/2021/451032021

A Unified Framework for Encryption and Decryption of Images Based on Autoencoder (UFED)

Muhammad Azeem[1], Dr. Shazia Saqib[2], Farrukh Sadiq[3]
[1] Lahore Garrison University (LGU), Pakistan, azeemzulfiqarcs@gmail.com
[2] Lahore Garrison University (LGU), Pakistan, Shaziasaqib@lgu.edu.pk
[3] Lahore Garrison University (LGU), Pakistan, Farrukhsadiq6@gmail.com

ABSTRACT

Over the last decade, many disciplines have made great strides in deep learning technologies, especially in computer vision and image processing. However, video coding based on deep training is still in its initial stage. This research work discusses the representative's work on deep learning for image/video coding, a research area since 2015. With the number of devices increasing on the Internet, we face low-cost transmission over a network and security and safety. We can't determine the accurate data size with encryption and decryption cost and amount of noise in communication. Our proposed unified framework for encryption and decryption of images based on an autoencoder (UFED) can control the cost during encryption and decryption using modern techniques like deep learning and neural network. The Autoencoder is worked as close to CNN and is trained on images and video frames to extract the image's feature. In this framework, the encoder changes the image into latent space or compressed form in a small size. We achieved the best image-compression ratio with Autoencoder over JPEG; JPEG typically achieves 10:1 compression with little perceptible loss in image quality. This research observed the accuracy of image reshaping from latent space as well. We have achieved over 97.8% accuracy on the standard quantity evaluation measure in our proposed deep learning technique, far better than previously implemented models.

Key words: Autoencoder; Artificial Neural Network; CNN; Multilayer Perceptron; Encoding-Decoding

1. INTRODUCTION

Quality of life has been an essence of various factors, and one of the fundamental factors is cost, security, and safety. With the immense increase in social interactions through internet devices like IoT, we face low-cost transmission over a network and security and safety. In other frameworks for detecting noise and decryption and encryption, we cannot determine the accurate data size with encryption and decryption cost and amount of noise in communication. On the other hand, in a unified framework for encryption and decryption of images based on an autoencoder, we can determine the transmission noise. We can control the cost during encryption and decryption by using modern techniques like deep learning, auto-encoders, and neural network; by using these approaches, we can control the noise in the transmission, and the main thing to secure the data we can control the cost of the data using the autoencoder. This framework is fast reliable and, provides the maximum accuracy to detect the noise and cost of the transmission, and provides a secure encryption method and decryption of data.
With the improvement of mixed media innovation and communication innovation, sight and sound diversion has assumed a significant job in individuals' everyday lives. Pictures and recordings take up the primary piece of sight and good amusement. It carries grim tests to store and transmit that information and advances higher prerequisites on the constrained transfer speed web, particularly for enormous and high-caliber computerized pictures. The Internet of Things (IoT) has undeniable influence (IoT), IoT devices are now in 60 percent of homes in developing countries that have access to the internet. By 2021, an estimated 200 million automobiles will be linked to the Internet, with the potential to transform entire industries. By the end of 2019, approximately 26 billion devices are projected to be connected to the IoT. It will only expand as time moves on, as these intelligent devices create more data and as technology continues to advance. These numbers are expected to increase exponentially to 50 billion IoT devices communicating by 2021 and as many as 75.44 billion IoT devices by 2025. However, this also concerns some concerns you should be aware of, including security concerns, reliability, and transactions' validity. With the number of devices increasing on the Internet, we are using different encryption and decryption techniques, which increase the cost of data and noise in communication.
Here are some key daily statistics [31]:

- 500 M tweet
- 294 B email
- 4 P.B. of data are created on Facebook
- 4 T.B. of data is generated from each connected car
- 65 B messages are sent on WhatsApp
- By 2025, it is estimated that 463 GB of data will be created each day globally – that is the equivalent of 212,765,957 DVDs per day!

1.1 Autoencoder vs. encryption and decryption of data

An Autoencoder is a technique; it converts the original information into latent space. So autoencoder is the combination of encoder and decoder. Still, their architecture and model can be different, so the autoencoder's function is to convert the images into latent space, and then it will be transferred and encoded again. Hence, this feature is to express in a minor vector, and the encoder gives us this vector. It then retrieves the information back from the vector. An autoencoder is a neural network technique consisting of three layers; the first input, the second encoding of the hidden layer, and the decoding layer. This technique is used to reconstruct the information into original data. Figure 1 shows an autoencoder is working on unsupervised learning.

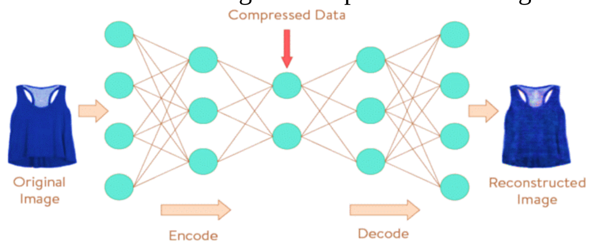


Figure 1: working of an autoencoder

Image/video encoding is a key and enabling advancement for deep image processing and computation based on visual. Imaginative work of Image/video coding backed to the 1960s, a long time before the ascent of modern image processing and communication systems. For example, the image Coding Symposium was moved in 1969, a prominent all-inclusive social occasion based explicitly on advances in Image/video coding. Starting now and into the foreseeable future, science and industry have advanced unfathomable endeavors here.

1.2.1. Image and Video Compression:

There is a lot of image/video compression framework and technique, but JPEG is the most common among them [1]. The image is broken down into 8x8 blocks, and Block-DCT is used to convert each block into the frequency. The entropy coding is used to convert each block into a binary stream. MPEG-2, H.264/A.V.C., and HEVC are the most common and popular video coding standards adopt for prediction-based coding. JPEG and HEVC are used to predict the neighboring reconstructed blocks.

Besides these technologies, a neural network is also used to code or predict the blocks [2].

1.2. Motivation:

In other frameworks for detecting noise and decryption and encryption, we cannot determine the accurate data size with encryption and decryption cost and amount of noise in communication.

On the other hand, in a unified framework for encryption and decryption of images based on an autoencoder, we can determine the transmission noise. We can control the cost during encryption and decryption using modern techniques like deep learning, autoencoder, and neural network. Using these approaches, we can control the noise in the transmission, and the main thing to secure the data we can control the cost of the data using the autoencoder. This framework is fast reliable and, provides the maximum accuracy to detect the noise and cost of the transmission, and provides a secure encryption method and decryption of data. In digital media, the cheap transmission of visual data is a very challenging and hot research topic. This research intends to propose autoencoder-based encryption and decryption of images, decreasing the cost of transmission of visual data. After studying the existing techniques, there is still a need to develop data cost-effective cryptography and noise detection techniques.

This paper proposed a unified framework that will overcome the drawbacks of these techniques, and this proposed unified framework is more efficient, cost-effective, and secure.

2. RELATED WORK

This chapter summarizes the work done by other authors and researchers in the field of data compression. The chapter details how data compression helps the network make their data secure and low cost. Results of surveys are also discussed in this chapter that shows the increasing importance of data compression.

Balle et al. used combined I.D.D. (the noise for each pixel is independent and identically distributed) noise uniformly and pretended the quantizer while training the CNN, enabling S.G.D. (Stochastic Gradient Descent) method to meet streamlined demand. This technique complies with JPEG2000, as shown by both PSNR and MSSSIM measurements. Balle' and his partners have also extended this model by using scaled hyper to entropy estimates [3]. The performance and the objectives achieved with HEVC were similar to the previous technique. Minnen et al. and others further improved a model called entropy coding for image compression [4] and outdid the intra coding for HEVC. Support and the vitality efficiency investigation should be investigated since the autoregressive segment is not effectively parallelizable. More improvement in image compression was also made by Zhou et al. by using a fusion structure with pyramidal characteristics in the encoder end and a post-processing filter based on CNN in the decoder end [5]. Entropy coding and quantization can be used jointly for image compression in [6][7] and image compression based on CNN prediction in [8].Li et al. developed a compacted illustration model based on CNN to advance the high pixeled image compression framework established on CNN by limiting the data loss of low pixel-based images [9]. Yang et al. proposed a neural network to improve the quality of multiple images for compressed videos using adjacent high-quality images to enhance low-quality images. A machine-based detector with a carrier vector is used to search for high-quality images in the compressed or low-quality video [10].CNN-based feature improvement moreover offers compelling quality in multicast and in-depth video encoding. Zhu and other CNN models considered for after-processing of displays made to improve the performance of 3-dimensional video encoding [11]. Further work has used a more complex structure to improve compressed images [12] [13].Chen et al. suggest a study for the recipe of some CNN networks, named as DeepCoder, which reached the same superiority low profile x264 encoder [14]. The intra expectation is executed through a CNN system to produce an element map, indicated as fMap. The entomb forecast is obtained from movement estimation on past edges produced in DeepCoder. The fMap is additionally encoded and quantized inflow. The intra-and between expectation residuals are changed into an increasingly reduced area utilizing neural systems. The procedure of that is comparable with fMap age in intra forecast yet with various neural parameters. The method used by the two entity residual and intra forecast is coded and quantized by using entropy coding of Huffman. Even though there is no the same number of coding apparatuses as H.264, DeepCoder illustrations practically identical compression execution contrasted and H.264, which illustrations another video coding answer. In 2018 Chen et al. suggested a completely video-coding structure based on learning via presenting the idea of Voxel CNN by means of investigating spatial-fleeting rationality to perform prescient coding inside learning system [15] viably. HEVC is the best video coding standard in class, accomplishes the requirements to control the cost of distortion and compression performance. By leaving the pointless R.D. calculations, the cost for computational can be controlled. The quick mode-choice calculations are suggested for C.U. (Coding Unit) and P.U. (Prediction Unit) that base on neural network, that are equally inviting as well as simple for VLSI structure[16],[17].

In particular, the fast algorithm analyzes local gradient bases to categorize blocks into edge categories and homogeneous. This technique will overcome the burden of CNN and evade CNN's uncomplimentary condition due to homogenous barriers. So, CNN is formed such that edge-blocks overcome at least two coding unit separation modes in every C.T.U. for the entire optimized procedure for distortion. A substantial network level is designed with a maximum pool level, followed by three fully integrated levels. It brings together the Q.P. values in the neural network at the last fully integrated level. Each quadrangular C.U. is cast-off as a neural network input, and the product is the bool decision for quads or no divisions for the present C.U. This eliminates the requirement to edit and select the recursive method. According to this method, it saves 61.1% time for coding, and the BD-rate loss is just 2.67% with respect to HM-12. Xu et al. gave a case study and suggested the whole C.T.U. Breakage structure by using LSTM and CNN to find if the decision for mood stopped prematurely [18].

We sought to develop an innovative framework for the visual signal display that smartly supports human vision display and image processing analysis for system-supported image and video compression. Given the low cost and significance of extracted features for optical semantic descriptions, such as CNN properties, we suggested the ordered visual illustration of signal in [19] through combined compression of features descriptions and visualization content. For each video image frame, descriptors extracted feature and then compressed, after this decoding feature activated and control the large scale visual compression of data through motion. This procedure aims not to compress the data but also to ensure analytical efficiency by erasing the video's functionality without compromising the compressed objects. Using [20], model based on CNN for video and image compression, the model used for compression is a multivariable optimization problem that works together, taking into account processing costs, the rates, and CNN performance used for CNN transmission (if necessary) to optimize. Summary of problems that arise in VANETs [32].

Earlier efforts [21] had suggested a formulation to optimize the distortion of complexity under performance constraints for the video encoding problems, which can be expanded to optimize the CNN model's compression with the computation cost and video performance compression. In November 2019, the article presented an architecture that works based on the D.C.T. and compressed the JPEG image with the help of VHDL [22].

3. CONVOLUTIONAL NEURAL NETWORKS (CNN)

In the 1950s, Frank Rosenblatt named scientist introduced the concept of a perceptron with the help of earlier work prosed by McCulloch and Walter [23]. With the increase in applications of machine learning algorithms and the advancement of technology-led towards the classification of image data. In the start, each pixel of the image was taken as separate input just like numeric data. This technique's problem was that all the pixels were considered individually, which resulted in equal treatment of pixels close or farther to each other. There was a need to extract spatial features, features with respect to position and grouping of pixels, from image data for better classification. In image classification problems, for extraction of spatial features of an image, convolutional neural networks (CNN) were introduced [2]. In CNN's a fixed/variable size of the window, also known as a filter, is convolved on the whole image to extract features from the image. This filter treats a group of image pixels as an input and generates a single value from all these pixels [24]. More than one filter of different size is applied to the input image to extract different and adequate features from images. The most frequently used filter sizes in CNNs are 33, 55, 77, and 99. So for a single convolution on an image, a filter is placed at the top left corner of the image and moved one step further towards the right side until the end of the image. After this, the filter is moved one step down from the previous convolution and again moved towards the image's right side. This process yield and a new set of pixels or image, containing different features of an image. This process includes one step forward and downwards jump of the filter while in some cases, this jump can be of 2 or 3 pixels, and jump size is called stride. This kind of multiple convolutions is done in a single layer that yields multiples features/images convolved in further layers. These filters are the weights used to extract features from images and are updated by any optimization algorithm during training to extract important features. Initial level convolutions extract low-level features like edges and point detection. At the same time, later layers convolutions extract high-level features like objects and shapes in input images. In modern convolutional networks, pooling layers are also used along with convolutions in the networks. Pooling layers are used to reduce the feature map size extracted from images using convolutional layers and are used in between the convolutional layers of the network. Mostly frequently used pooling layer is the 22 max-pooling layer in which the image is divided into a group of pixels having 4 pixels in each group. The pooling layer outputs the maximum value from each pixel group, hence reducing the feature map size. Another frequently used layer in convolutional neural networks is a softmax layer. Softmax layer is used as the last layer of CNNs, and it converts the outputs of neurons into 0 or 1 value. An example of CNN architecture with a 28x28 sized input image is shown in Figure. 2.5. The filter size for the convolutional layer is 55, and 20 filters are used for convolutions. These 20 filters yield 20 feature maps having a size of 24x24. In Figure architecture, the second layer is the max-pooling layer of size 2, which reduces the features map to half. The third layer of the CNN architecture is a fully-connected layer containing 100 neurons, which takes input from the previous layer. The output of the fully connected layer is used for the last layer having 10 neurons. 10 neurons in the last layer mean 10 output classes and the last layer is the softmax layer to convert the output of neurons into the probability of each class. Figure 2 shows layers of CNN.

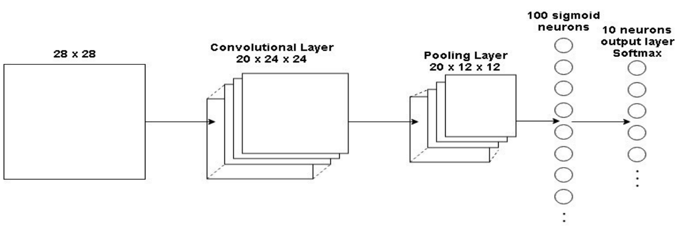


Figure 2: CNN layers

4. AUTOENCODER FRAMEWORK

A summary of Autoencoder and the types is available in this part of paper.

4.1 Overview: An autoencoder has three main elements: encoder, code (compression layer), and decoder. The training method resembles a training feed forward neural network through back propagation [25]. Autoencoder can be supposed as lossy, meaning the output data will low down the first input signal with respect to nature. The main objective of the signal is to overcome the reconstruction error or loss of data. That idea beyond autoencoders is the minor possible distortion of the input to the output[26]. This can seem unnecessary; however, the expectation is that valuable qualities of this data are drawn in the training duration. Figure 3 shows autoencoder architecture.

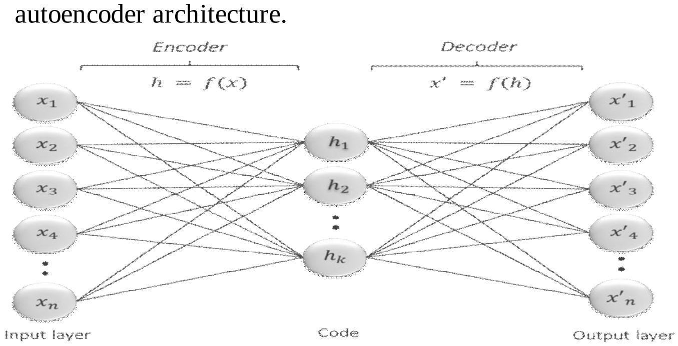


Figure 3: Autoencoder architecture

4.2 Autoencoder algorithm

A Realistic mapping called the encoder allows the input and first maps to be interpreted as a secret representative of code. This can be used as the encoder: A parameter,∈ □^{n} Each shows the autoencoder input and size n equal to the input size

□ $= f_\theta(x) = \sigma(xW + b)$, (4.1)

Where the weight matrix is $\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{b}\}$; bias vector is $\boldsymbol{W} \in$ □^{mxn}; $\boldsymbol{b} \in$ □; $(\boldsymbol{x})$ – the encoder; $\boldsymbol{\sigma}$ is an activation function.

The secret image is linked primarily with an output layer decoder,∈ □^{n} in just the same n-dimension as the origin. This procedure can be determined by the equation:

$x' = g_{\theta'}($□$) = \sigma($□$W' + b')$, (3.2)

Where $\boldsymbol{\theta}' = \{\boldsymbol{W}', \boldsymbol{b}'\}$; $\boldsymbol{g}_{\theta'}(\boldsymbol{h})$ – the decoder.

Another way of restricting the W decoder's weight matrix is to convert the $W' = WT$ embedding weights. This is known as linked masses, use of linked masses is possible, or even some unlinked weight trials have shown comparable findings [27].

4.3 Parameterization of Autoencoder

To start or prepare the autoencoder, some main parameters must be pre-defined. Pre-defined or Pre-set parameters will define the autoencoder architecture like the number of layers size of the code.

4.3.1.1 Under complete Autoencoders

In an under complete autoencoder size of the input always greater than of input size. These autoencoders are intended to catch helpful features in the information and diminish its dimensionality by speaking to the entire populace with detected silent features. For this situation, the system is urged to become familiar with compression.

4.3.1.2 Overcomplete Autoencoders

The dimensionality reduction debate is developed on the premise that the code size is bigger than the input. However, despite all the contrary announcements, an autoencoder can acquire some helpful information regarding the structure by applying some restrictions on the action of these hidden representations [28]. After the structure is suitably corrected with the addition of the regularization conditions, the autoencoder benefits other attributes; moreover, the autoencoder can convert the input to a recreate able to output.

4.3.3 Loss function

The objective function assesses the efficacy of neural networks. This shows how the neural network is performing with great accuracy. In autoencoders, measures such as rebuilding for the significant reason that the expression reduction function, the role is much more satisfying. Here, the widely used loss functions will be considered, even though the option of functions is a great deal richer.

4.3.4 Optimizer

And those are the equations that attempt to pick the ideal mathematical function values used in computer program testing. Most of the optimization algorithms are gradient descent algorithms. The gradient descent algorithm varies according to how much data the gradient is changed, and the gradient descent expansion depends on various methods of adjusting the training set and selection of model parameters. In this research, we have used Adam Optimizer.

4.3.5 Regularization

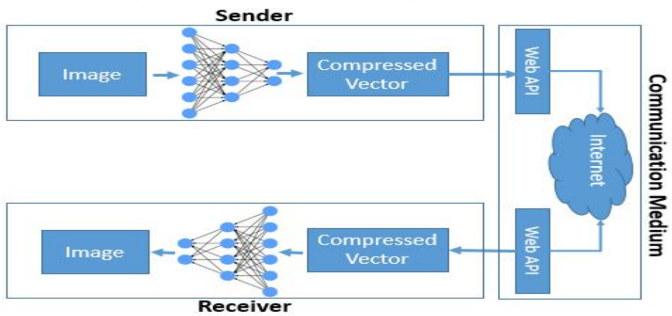
At the end of the discussion, improving the algorithm's efficiency is said to a regularization, not only performance but also on the new data. A particular regularization method modifies the algorithm with the addition of some constraints. These subsections describe regularizers that apply to autoencoders.

5. METHODOLOGY

This section of the paper presents significant phases of the research cycle and depicts how implementation was led. It gives a complete description of the dataset, starting information comprehension through an exploratory examination of its factors, includes extraction of features and their representation or visualization, the process of training of data, and at the end, evaluation or assessment strategies.

5.1 Proposed Methodology

The proposed system consists of three main modules. The first sender, the second communication medium, and the last is the receiver. Figure 5 shows block diagram of the proposed



methodology.

Figure 5: Block diagram of the proposed methodology

The details of the above mentioned three modules are given below:

Sender:

Image: A source image or video is given to the system transmitted over a network.

Encoder: The encoder will extract some features and convert them into a latent space or compressed/semantic vector.

Compressed vector: It consists of the compressed form of the data.

Communication Medium:

Web API: Django framework (pre/post method)

Internet: Any connection or medium

Receiver:

- Compressed vector: It consists of the compressed form of the data.
- Decoder: It will decode again into its original form.
- Image: At last, the final compressed image or video with compressed size will reappear.

5.2 Extracting features with autoencoder

The numbers of characteristics describing the sample must be calculated in unchecked education from a high to a low-dimensional field. Simply specify the number of secret units in the mid-layer of the autoencoder architecture. Unlike the textbooks, there is no such simple rule while projecting the actual information using an automobile encoder to determine a small-dimensional space. Some literature proposes thumb rules for choosing several secret units in a neural system (Blom, 1991; Swenglar, 1995). Many articles provide scientific support methods with mathematical proof (Xeo & Chan, 2009). In this analysis, the number of troops concealed was generally defined based on a determination on the range of values required by testing different extraction algorithms to determine 70-90 percent (Bugar & Gotarmen, 1998) and preserve equity the input measurements. Figure 6 shows proposed methodology.

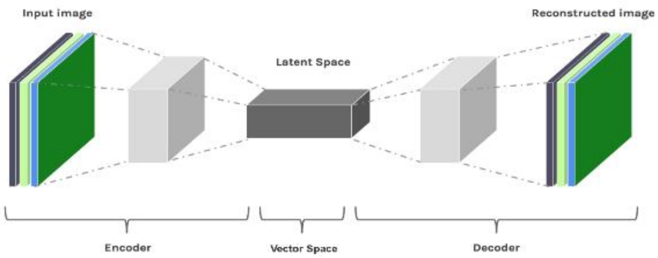


Figure 6: Diagram of the proposed methodology

6. RESULTS AND DISCUSSIONS

6.1 Dataset

Deep learning technology is being chosen for the problem. It is a data-hungry technique. Large data is required to train this algorithm to apply. Secondly, the data used should be accurate and preprocessed, and it should be relevant to the problem. Collecting the correct data and preprocess are critical features in this kind of research. There are online repositories available that have data sets for different problems. We choose the following from the internet repository:

6.1.1 MNIST

Most of the experiments are performed on the MNIST dataset containing 60000 training and 10000 testing images of handwritten digits. Digits in the training set were written by 250 writers, assuring digits in the test set were written by distinct writers. Images stored in this database have a resolution of 28x28 pixels and contain intensity values in grayscale format, represented by 8 bits per pixel, resulting in 256 intensity levels. MNIST dataset has been selected due to its availability and popularity, being the most popular benchmarking dataset of handwritten digits. Figure 7 shows Results of MNIST dataset.

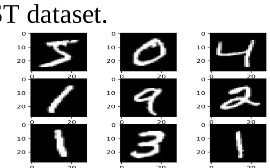


Figure 7: Results of MNIST dataset

6.1.2 CIFAR-10

In some of the computationally inexpensive experiments, D.N.N.'s robustness when classifying the CIFAR10 dataset was tested. Dataset is partitioned to the training set of 50000. Design 22 images and a testing collection of 10000 images. In total, there are 10 classes of 6000 images per class. CIFAR10 classes are aeroplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Every image consists of 32x32 pixels expressed by the RGB value. Each pixel's color channel information is represented by 8 bits. We use two main subs of the 80 million vector graphics database in this study. This dataset consists of 80 million smaller images, and 32 or 32 color images searched for both non-abstract English names and nine phrases of the WordNet lexical database using various online image search engines. This data collection displays a random image sample in Figure.5.4. There is a very noisy mark in any image in the dataset. These loud stickers are not used in any way. We use an unlabeled subset of 1.6 million pictures for unchecked pre-training. Figure 8 shows results of CIFAR-10 dataset

Figure 8: Results of CIFAR-10 dataset

6.2 Results

The objective function assesses the efficacy of neural networks. This shows how the neural network is performing with great accuracy. In autoencoders, measures such as rebuilding for the significant reason that the expression reduction function, the role is much more satisfying. Here, many hidden layers are taken into account, even though the choice of functions is much richer. The loss function curve stabilizes from all diagrams after approximately five thousand times, and the remaining elements will not decrease dramatically. Every optimizer in Keras has a default learning rate for each training session. One of the essential aims of this investigation is to reduce the loss mechanism. The graph figure 9 illustrates the loss of instruction and validation.

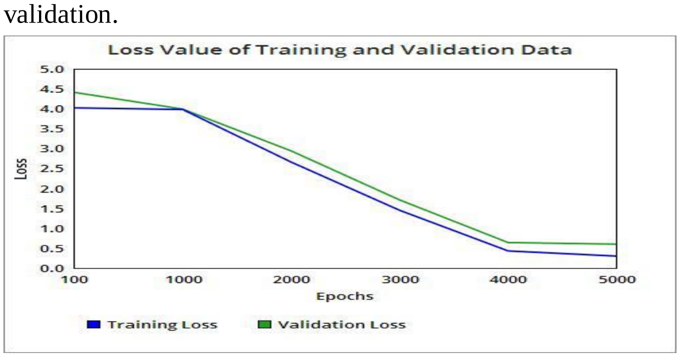
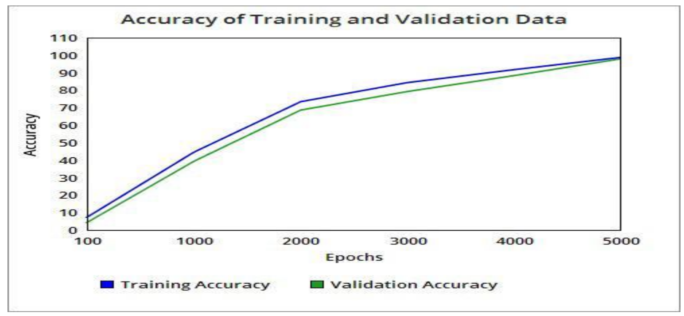


Figure 9: Accuracy of Training and Validation Data

The graph shows that when training is the start, training and validation are at their highest value. However, as the training continues, this loss decreases. At 1000 epochs, this loss decreases significantly. This graph shows that as the training goes on, loss decreases. As loss decreases, accuracy increases. In autoencoder, the main focus is that the training image has a targeted object and background. It is necessary to pass the detected object to the model for the feature extraction of the object.

6.3 Training and validation Accuracy

The following graph shows training and validation accuracy. As training starts, accuracy is low. This is because the loss is high, and the system has just started the training. But as time increases, accuracy is also increased. The maximum accuracy obtained at 300000 epochs is 97.8. Figure 10 shows Accuracy of Training and Validation Data



Activation function	sigmoid
Optimizer	adam
Loss function	Binary cross-entropy
Learning rate	0.001
Epochs	5000
Normalization method	MinMax

Figure 10: Accuracy of Training and Validation Data

6.4 Comparison with other architecture

Shawna A et al. presented an architecture that works based on the D.C.T. and compressed the JPEG image using Very High Speed Integrated Circuit Hardware Description Language (VHDL) [22]. The result of the compression is Table 1.

Table 1: Image Comparison performance

Image (jpeg)	Dimension	Original Size (KB)	Compressed Size (KB)	Reduction (Percent)
Desert	1024*768	846	127	84.98%
Koala	1024*768	781	160	79.51%
Lighthouse	1024*768	561	100	82.17%
Penguins	1024*768	778	119	84.70%
Tulips	1024*768	621	96	84.54%

As in study suggested, we reached the highest resolution level with both the automotive encoder across JPEG; JPEG compares favourably 10:1 for low visual loss. JPEG is by far the most common image compression format worldwide since its launch in 1992 [29]. Results are in table 2.

Table 2: Image Comparison with different datasets

Image (JPEG)	Technology	Dimensions	Original Size	Compressed Size	Reduction %
CIFAR-10	Autoencoder	32x32	1024	81.92	92%
Desert	JPEG	1024x768	846	127	84.98%
Koala	JPEG	1024x768	781	160	79.51%

On the other hand, we observed the accuracy of image reshaping from latent space as well. Results are being compared with other models in terms of accuracy. Table 3 show the accuracy table.

Table 3: This paper Accuracy comparison with other Algo

Author	Technology	Accuracy	Dataset
Own	Autoencoder	97.8%	CIFAR-10&MINST
VGG-D [30]	ConvNet	92.8% (test)	ImageNet

This data shows that accuracy with autoencoder is high as compared to other algorithms. For this reason, autoencoders have been used.

7. CONCLUSION AND FUTURE SCOPE

We cannot determine the accurate data size with encryption and decryption cost in other frameworks for detecting noise and decryption and encryption.

We have proposed a novel deep UFED framework for image transmission over a communication network. The autoencoder is worked as close to CNN and is trained on images and video frames to extract the image's feature. In this framework, the encoder changes the image into latent space or compressed form in a small size. The latent space or vector form transmit over a communication channel, and then it converts back to an image from vector form. The latent space contains all the features that are enough to reconstruct an image from a vector. We compared the compression ratio with JPEG that achieves an image with noise. This framework is fast reliable and, provides the maximum accuracy to detect the noise and cost of the transmission, and provides a secure encryption method and decryption of data. In the future, we shall improve data transmission performance and check the extensive dimensional data. In the next step, we shall also work on the channel codes and design the algorithm to cover the channel codec. We aim to implement this process as GAN-based in the future. In addition, we can look at the device output for non-Gaussian streams and memory channels; for that, we have no stream code solution. In this non-ideal setting, we hope that the advantages of the NN-based UFED system would be more apparent.

REFERENCES

1. Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, **Deep learning for visual understanding: A review,** Neurocomputing, vol. 187, pp. 27–48, 2016.
2. S. Loussaief and A. Abdelkrim, **Deep learning vs. bag of features in machine learning for image classification** in 2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET), 2018, pp. 6–10.
3. J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, **Variational image compression with a scale hyperprior.** arXiv Prepr. arXiv1802.01436, 2018.
4. D. Minnen, J. Ballé, and G. Toderici, **Joint autoregressive and hierarchical priors for learned image compression.** arXiv Prepr. arXiv1809.02736, 2018.
5. L. Zhou, C. Cai, Y. Gao, S. Su, and J. Wu, **Variational autoencoder for low bit-rate image compression.** in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 2617–2620.
6. E. Agustsson et al., **Soft-to-hard vector quantization for end-to-end learning compressible representations.** arXiv Prepr. arXiv1704.00648, 2017.
7. L. Theis, W. Shi, A. Cunningham, and F. Huszár, **Lossy image compression with compressive autoencoders.** arXiv Prepr. arXiv1703.00395, 2017.
8. E. Ahanonu, **Lossless image compression using reversible integer wavelet transforms and convolutional neural networks.** 2018.
9. Y. Li, D. Liu, H. Li, L. Li, Z. Li, and F. Wu, **Learning a convolutional neural network for image compact-resolution.** IEEE Trans. Image Process., vol. 28, no. 3, pp. 1092–1107, 2018.
10. R. Yang, M. Xu, Z. Wang, and T. Li, **Multi-frame quality enhancement for compressed video.** in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6664–6673.
11. L. Zhu, Y. Zhang, S. Wang, H. Yuan, S. Kwong, and H. H.-S. Ip, **Convolutional neural network-based synthesized view quality enhancement for 3D video coding.** IEEE Trans. Image Process., vol. 27, no. 11, pp. 5365–5377, 2018.
12. L. Cavigelli, P. Hager, and L. Benini, **CAS-CNN: A deep convolutional neural network for image compression artifact suppression.** in 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 752–759.
13. B. Zheng, R. Sun, X. Tian, and Y. Chen, **S-Net: a scalable convolutional neural network for JPEG compression artifact reduction.** J. Electron. Imaging, vol. 27, no. 4, p. 43037, 2018.

14. T. Chen, H. Liu, Q. Shen, T. Yue, X. Cao, and Z. Ma, **Deepcoder: A deep neural network based video compression**. in 2017 IEEE Visual Communications and Image Processing (VCIP), 2017, pp. 1–4.
15. Z. Chen, T. He, X. Jin, and F. Wu, **Learning for video compression.** IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 2, pp. 566–576, 2019.
16. Z. Liu, X. Yu, Y. Gao, S. Chen, X. Ji, and D. Wang, **CU partition mode decision for HEVC hardwired intra encoder using convolution neural network.** IEEE Trans. Image Process., vol. 25, no. 11, pp. 5088–5103, 2016.
17. N. Song, Z. Liu, X. Ji, and D. Wang, **CNN oriented fast PU mode decision for HEVC hardwired intra encoder**. in 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2017, pp. 239–243.
18. M. Xu, T. Li, Z. Wang, X. Deng, R. Yang, and Z. Guan, **Reducing complexity of HEVC: A deep learning approach.** IEEE Trans. Image Process., vol. 27, no. 10, pp. 5044–5059, 2018.
19. X. Zhang, S. Ma, S. Wang, X. Zhang, H. Sun, and W. Gao,**A joint compression scheme of video feature descriptors and visual content**. IEEE Trans. Image Process., vol. 26, no. 2, pp. 633–647, 2016.
20. Y. Li et al., **Joint rate-distortion optimization for simultaneous texture and deep feature compression of facial images.** in 2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM), 2018, pp. 1–5.
21. L. Su, Y. Lu, F. Wu, S. Li, and W. Gao, **Complexity-constrained H. 264 video encoding.** IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 4, pp. 477–490, 2009.
22. A. Shawahna, M. Haque, and A. Amin, **JPEG Image Compression using the Discrete Cosine Transform: An Overview, Applications, and Hardware Implementatio.** arXiv Prepr. arXiv1912.10789, 2019.
23. M. A. Nielsen, **Neural networks and deep learning**, vol. 25. Determination press San Francisco, CA, 2015.
24. A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, **Deep learning for computer vision: A brief review.** Comput. Intell. Neurosci., vol. 2018, 2018.
25. P. Baldi, **Autoencoders, unsupervised learning, and deep architectures.** in Proceedings of ICML workshop on unsupervised and transfer learning, 2012, pp. 37–49.
26. I. Goodfellow and Y. Bengio, **Courville. A (2016) Deep learning.** MIT Press. available), 2017.
27. D. Erhan, A. Courville, Y. Bengio, and P. Vincent, **Why does unsupervised pre-training help deep learning?** in Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 201–208.
28. A. Ng, “Sparse autoencoder,” 2011. .
29. G. Hudson, A. Léger, B. Niss, I. Sebestyén, and J. Vaaben, **JPEG-1 standard 25 years: past, present, and future reasons for a success.** J. Electron. Imaging, vol. 27, no. 4, p. 40901, 2018.
30. K. Simonyan and A. Zisserman, **Very deep convolutional networks for large-scale image recognition.** arXiv Prepr. arXiv1409.1556, 2014.
31. Jeff Desjardins, Apr 17, 2019. Accessed on:Nov.1,2019. [Online].Available: https://www.weforum.org/agenda/2019/04/how-much-data-is-generated-each-day-cf4bddf29f/
32. A.Sajithabegam, Karan V , Nithish Kumar G, Venkatesh Prasad D, **Cloud Based Approach to Enhance Security in VANET Communication.** International Journal of Advanced Trends in Computer Science and Engineering Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse801022021.pdf
https://doi.org/10.30534/ijatcse/2021/80102202