Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7923001

INTERPOLATIVE MODEL ON HEURISTIC PROJECTION TRANSFORM FOR IMAGE COMPRESSION IN CLOUD SERVICES

[1]R. Pushpalatha, [2]Ramchand Kolasani

[1]Department of CSE Engineering ANU College of Engineering and Technology, Guntur, AP

[2] Principal, ASN Degree College, Tenali, Guntur, AP

ABSTRACT:

Image compression analysis have become most prominent feature to realizeto observe, process and store image and its features in local or cloud storages. Since the trend of ML and DL have become the important features that are effective in real time design, we tend to observe the loss and its data manipulation via manual or automatise the compression standard based on the applications. Since, storing of data in less memory stricture and its architecture have become the recent interests of the compression standard with its time features. Presently, we have observed most of the design are either hybrid or novel structures with its transformative analysis to reduce the performance parametric with less memory storage and also less time consumption. But the prominent change of the Image compression rate with its transformative approaches have depicted more loss in image while compression ration increases with pixel quality. So, as to provide such change in the image compression our proposed model implemented a novel intuitive orthogonal Transformation and Heuristic Projection Orthogonal Transformindicating loss less than 1% observed with HPOT-Dense-NN. The effective parametric with different image dataset such as CIFAR-10, MNIST and 100 sample real time datasets are considered to implicate the overall comparison with State of Arts (JPEG, JPEG-2000, GAN, LSTM and CNN).The overall bitrate, compression rate and PSNR with SSIM are implemented with the proposed design algorithms which have shown improved values better than SOA as tabulated.

Keywords: Image Compression (IC), Dense neural Net (DNN), multi-layer perceptron (MLP), Convolutional Neural Net (CNN), Deep Neural Net (DNN), Machine Learning (ML), Deep learning (DL), Orthogonal Transforms (OT),Heuristic Projection Orthogonal Transform (HPOT).

1. INTRODUCTION

In this day and age of massive amounts of data, the quantity of the data itself is the primary worry of academics and experts. Data compression is necessary for the proper transmission and storage of data due to the restricted bandwidth of the channel and the limited space available in memory [1]. This makes it necessary to compress data in order to ensure that important information is not lost. Audio compression, picture compression, video compression, and document compression are only few of the different types of data compression that can be accomplished [2]. An image may contain information that is either valuable, redundant, or irrelevant depending on how it is interpreted. For the purpose of image compression, it is possible to disregard information that is not important. In order to bring out the details in photographs, it is essential to include redundant information, whereas information that is valuable is neither redundant nor irrelevant. We are unable to adequately rebuild or decompress images because we do not have all of the necessary information [3]. There are two primary groups to consider when discussing image compression. The first method is called lossless image compression, and it doesn't remove any of the original image's information. The second method is called lossy image compression. Lossless picture compression algorithms are particularly efficient for small-size data. Lossless approaches that are efficient for tiny data include Huffman coding, run-length encoding (RLE), arithmetic coding, Lempel-Ziv-Welch (LZW) Coding, and JPEG-LS [4]- [5]. The most significant disadvantage of the lossless compression techniques is that they are far less efficient at compression than the lossy compression approaches. For this reason, a large number of academics are focusing their attention on the compression of images using ML. There are a lot of studies that are concentrated on picture compression. Conventional image compression methods based on discrete cosine transform (DCT) and discrete wavelet transforms have been the subject of a number of studies that examine both their advantages and disadvantages (DWT). The prediction and transform-based picture compression techniques were analyzed and surveyed by in [5]. The importance of prediction and transform-based conventional methods for image compression was underlined in this survey, as well as the applications of these techniques. Edge detection, gradient analysis, and block-based prediction were the foundations on which these prediction-based algorithms were built. Wavelets, on the other hand, formed the foundation of transform-based algorithms. Throughout the survey, both comparative and in-depth analyses of entropy coding were presented. On the other hand, this did not include a discussion regarding end-to-end based image compression architectures that made use of ML. The authors of [6] looked at DCT- and DWT-based algorithms for their research. The survey did not address the recently discovered methods of image compression. Similarly, the authors of [7] analyzed a variety of lossy and lossless picture compression techniques. The essay discussed the benefits and downsides of three different types of picture compression frameworks: predictive, entropy coding, and discrete Fourier transform-based image compression. The survey did not cover any aspects of machine learning-based image compression architectures. The traditional DCT, DWT, and entropy-based techniques to image compression have been highlighted as being quite important in a number of recent surveys.

Convolving using random filter kernels yields a large number of features, but many of these features are identical. As a result, the network's resources are being wasted by being used to repeatedly learn the same features. To add to that, certain features are superfluous and should not take up valuable processing time when training. Learning irrelevant information is undesirable for image reduction, since learning key features plays a critical role. There has to be a system in place to save the most informative bits of a feature while discarding the rest. The process should direct the attention of CNN layers onto the crucial parts of the image. In most cases, a deep CNN network would experience sluggish and insufficient training due to decreasing feature reuse [8]. Larger receptive fields take into account more distant pixels while calculating the final result. As a result, they choose for a more expansive network, which extracts numerous redundant characteristics and limits their capacity to generalize. Learning numerous duplicate features reduces the performance advantage of deepening or extending the network for compression- decompression applications. Overfitting occurs when a network incorrectly learns a prior by focusing on these duplicated characteristics, wasting computational and memory resources. Yet, in comparison to the depth and breadth, cardinality, which denotes the size of the set of transformations, decreases the total number of parameters and enhances the representation power.

1.1 Problem Statement

1.2 Objectives:

a) To implement a Hybrid, Transform approach with Orthogonality feature.

b) To realize a Deep and Machine learning models for Transformer approach based on Orthogonality

c) Improvise a feature extraction model with hybrid interpolative approach for Image compression analysis for image size reduction

d) Compare the SOA architectures for all performance metrics.

1.3 Overview:

The design on Image compression, has realized the approach on orthogonality and its importance feature extractions, and transform approaches with introduction part. While in section -2 the work depicts the overall survey of different architectures of Image compression with ML and DL. The existing model with hybrid transformer approaches and existing orthogonality features are described to realize the gap and current research problems. In section -4 we have implicates four objectives indicating the current research problem and its analysis with effective weight prediction model based on Hybrid transform approach on Orthogonality. The proposed model on Deep learning effective reduces the overall compression loss by 1% for real time images with intuitive layer prediction weight feature. Finally, overall comparison with LSTM, Hybrid CNN, other GAN architectures with M-net, JPEG, JPEG-2000etc. are compared with proposed model indicating the structural difference.

2. LITERATURE SURVEY: 2023

In 2023, high-throughput, high-content, multispectral, and 3D imaging are all examples of cutting-edge microscopy methods that may generate hundreds of terabytes' worth of data each experiment. The efficient use of lossy image compression techniques, such as joint photographic experts' group (JPEG) and JPEG 2000, is essential for dealing with these massive data sets. While these techniques may provide good visual quality with large compression ratios, they cannot guarantee the integrity of medical data and information. Using colour wavelet difference reduction, this research suggests a new and superior way for compressing medical images. The suggested approach expands upon the conventional wavelet difference reduction (WDR) technique by using mean co-located pixel difference to choose the optimal number of colour pictures that offer the maximum similarity in the spatial and temporal domains. Images with high degrees of spatial and temporal coherence are compressed into a single volume and then analysed for things like PSNR and structural similarity index (SSIM). To test the efficacy of the suggested technique in the difficult area of histopathology microscope image analysis, 31 colorectal cancer slides are used. The visual quality of the medical picture is determined to be very good. The findings show that compared to JPEG 2000, the PSNR improvement might be as high as 22.65 dB. Our research has led us to develop a mobile and online platform for the purpose of compressing and transmitting real-time microscopic medical pictures, with compression gains of up to 10.33dB compared to a technique using discrete wavelet transform (DWT).

2022:

As point cloud is one of the most fundamental ways of expressing 3D environments and objects, point cloud compression (PCC) is an essential enabler for immersive multimedia applications. Octree-based Geometry-based Point Cloud Compression has seen several recent proposals for improving its average reconstruction quality (G- PCC). It is found, however, that when compared to G-PCC, these methods have a significant performance degradation in terms of point-to-point (D1) Hausdorff distance (octree). In this paper, we provide an almost lossless approach to compressing the geometry of a point cloud by using an adaptive residual compensation scheme that involves adding and deleting points depending on their defect severity. It retains a significant gain in average reconstruction performance over G-PCC while allowing for control of D1 Hausdorff (D1h) distance. Our approach has been tested on the solid point clouds from the MPEG Cat1A dataset, and it has been shown to be successful, with average bitrate reductions of 78.5% D1 and 11.4% D1h Bjontegaard-delta over the octree-based G-PCC.

Uses for point clouds nowadays range from the arts to autonomous vehicles. With the advent of these new uses comes the need of finding a way to compress the ever-increasing amounts of point cloud data in a way that is appropriate for both human display and machine processing. The goal of the JPEG Pleno Point Cloud activity is to establish a standard for point cloud coding that is based on learning, provides a single-stream, compact compressed domain representation, supports advanced flexible data access functionalities aimed at in-teractive human visualisation as well as effective performance for 3D processing and machine-related computer vision tasks, and so on. The JPEG Committee has been conducting a number of exploratory studies as part of this effort, with the goals of assessing the quality of currently implemented encoding standards, establishing reliable benchmarks, and investigating objective metrics against which novel, data-driven approaches to the problem can be evaluated. The essay introduces the reader to the JPEG Pleno Point Cloud project and then goes on to examine the difficulties of assessing and comparing other cloud coding methods, as well as some potential approaches to these issues. The JPEG Committee's methodology for assessing point cloud compression will be shown with experimental findings, and both the performance of state-of-the-art compression standards on point clouds and the sensitivity of the objective metrics used for this activity will be detailed.

The high spectral resolution of hyperspectral pictures makes them superior to standard RGB and multi-spectral photos for use in remote sensing. Images may be used to investigate phenomena seen rather than only verifying the existence of a certain feature. For both small and large satellites, the great spectral resolution presents a problem with data volume each observation. Time and power required for data downlinking is a major element in the latency of data and the imaging capabilities of a tiny earth observation satellite. Using satellite resources efficiently requires a well-thought-out image pipeline for acquiring, processing, and analysing observational data. Both in-flight and ground-based processing processes and technologies must be included into the imaging and processing pipeline. In this paper, we evaluate the speed and accuracy with which the compression ratio can be used to determine which

HYPSO-1 satellite observations should be prioritised for downlinking and which data sets seem to be of less value due to over/under exposure or cloud coverage, using a sample of the first observations from the satellite.

Point clouds generated by LiDAR in autonomous cars or robots may give more precise depth information of objects than 2D photos, but their vast amount of data makes them cumbersome for storage or transmission. In this work, we present R-PCC, a range image-based Point Cloud Compression technique that allows for either uniform or non- uniform accuracy loss during point cloud reconstruction. For the purposes of spatial redundancy and salient area categorization, we divide the original, massive point cloud into tiny, compact regions. Using a range picture, our technique preserves and aligns every point in the original point cloud inside the rebuilt point cloud, and the quantization module's settings limit the maximum reconstruction error. We demonstrate that our simpler FPS-based segmentation approach can outperform instance-based segmentation methods like DBSCAN in trials, and that our non-uniform compression framework significantly outperforms state-of-the-art large-scale point cloud compression methods in downstream tasks. As a reference point for range image-based point cloud compression, our real-time technique can achieve a compression ratio of 40 without compromising downstream operations.

Hyperspectral (HS) images have a high spectral information richness that may be used for a variety of purposes. The large amount of data included in an HS picture need compression before it can be stored or transmitted effectively. On the other hand, real-time compression is very difficult to achieve due to the complexity of the compression algorithms. In this paper, we offer a new approach for compressing hyperspectral images using a listless set partitioning. In order to compress data, the suggested approach makes advantage of inter and intra sub-band correlation in a zero-block cube tree structure. The findings demonstrate that the suggested compression algorithm has comparable coding efficiency to that of state-of-the-art methods while maintaining a low coding complexity. This means it has the potential to compete with other, more expensive hyperspectral image sensors.

In this study, we apply tensor analysis to the issue of signal and image processing. A brief overview of current techniques is provided within the context of the suggested approach; this covers both broad methods like princIOTl component analysis and independent component analysis and more narrow concerns like blind signal separation and picture compression. All of these techniques share the need of a singular value decomposition operation on the input data, which necessitates the use of a matrix or a succession of matrices to organise the information. Data may be presented concisely by cutting out unnecessary parts of the fundamental matrix format. Assuming that the data or signal sets are statistically independent, the results of a joint diagonalization of a succession of matrices are often adequate. Two examples are used to examine the supplied data and evaluate the value of the proposed method.

High-performance computing (HPC) systems are becoming larger and more complicated; thus, they often incorporate data-compression methods to cut down on data volume and processing time. Nevertheless, without locating the sweet spot between compression ratio and data loss, the advantages of lossy compression over lossless compression begin to erode. Several lossy compression methods exist, however transform-based lossy algorithms make better use of spatial redundancy. Since its compression performance on scientific data sets is not well understood, the transform-based lossy compressor has gotten less attention. The key takeaway from this research is that measuring dominating coefficients at the block level exposes the correct balance in transform-based lossy compressors, which may affect compression ratios as a whole. This motivates us to use statistical features that capture data properties in order to define three alternative transformation-based lossy compression algorithms with distinct information compacting approaches. Thereafter, we construct many prediction models using the statistical traits and characteristics of dominating coefficients, and we assess the performance of each model utilising six HPC datasets derived from three large-scale production simulations. Based on our findings, the random forest classifier accurately captures the behaviour of dominating coefficients with an impressive prediction accuracy of about 99%. Navigation in autonomous cars is very dependent on their perception system. When it comes to comprehending the environment around a vehicle, semantic segmentation far surpasses object detection. Nevertheless, constructing a real-time perception system based on semantic segmentation is made more difficult by the fact that comprehensive picture segmentation requires more processing resources than object identification. The limitations of perception systems are not only due to the nature of the underlying deep learning model. Transferring data, even over a fast network like Ethernet, comes with some delay. In order to make decisions about driving autonomously, the video feed from a vehicle must be sent to an edge device for processing. This research looks at the impact compression has

on the precision of semantic segmentation and how it might be used to reduce the transmission time of images at different JPEG compression settings. Although most studies on autonomous driving have been conducted in urban settings, we also want to utilise the Rellis-3D dataset to learn more about autonomous unmanned ground vehicles (UGVs) in the off-road domain. SwiftNet is a cutting-edge semantic segmentation model, and we determine its accuracy after training it on images with varying degrees of JPEG compression. Three photos are used to compare the transfer times of the various compression ratios. The results demonstrate a linear decline in precision with increasing compression ratios. SwiftNet with the greatest compression ratio of 16.96 obtains 67.9% mIoU when trained on the train set without compression, whereas the baseline gets 78.9% mIoU. By training SwiftNet on the relevant compression ratios, the accuracy of the higher compression ratios improves; the maximum compression ratio approaches 74.9% mIoU. Therefore, in all transfer cases, we see a favourable transfer speedup of these larger compression ratios when inducing JPEG compression: (a) 1870 pictures Picture (a), Video (b), and Picture (c) are all wrong. There is a 1.18x, 1.14x, and 1.06x acceleration in each case.

Improving compression algorithms for multimedia files is a major focus of academic and corporate research over the last several decades. Compression ratio improvements have lagged far behind the exponential expansion of picture, audio, and video data, so there will always be a trade-off between size and quality. We want to accomplish data compression via the use of neural networks and believe that multilayer neural networks provide a more effective option. In this research, we replace the standard transformations with a lossy compression architecture that takes use of the benefits of convolutional autoencoder (CAE). The experimental findings show that compared to the conventional coding techniques, our approach achieves higher compression ratios.

Compressing data before storing or sending it is a rapidly growing industry. With lossless compression, the compressed data does not need to be rebuilt precisely, but in lossy compression, data cannot be restored in their entirety. Compressing binary files, telemetry data, and high-fidelity medical and scientific photographs is a good example of a task for which lossless data compression is ideal. The optimal compression ratio cannot be guaranteed for each given data pattern by a single, universal approach. In this study, we offer a hybrid lossless hardware architecture that can compress a wide variety of data patterns without introducing any noticeable quality loss, including pictures, Gaussian distribution data, and repetitive data. It is recommended to profile data before compressing it in order to better choose suitable compression hardware. The suggested architecture is highly parallelized, allowing for compression and decompression of 64 bytes per cycle with little overhead. In addition, it offers a good compression ratio for both small and big blocks.

The extensive information stored in the successive frames makes it feasible to do a variety of post-capture actions (such as super resolution and HDR enhancement) that would not be possible with a single picture. As there will be more frames overall, it is crucial that these bursts be compressed well. To facilitate many downstream picture enhancing activities while simultaneously decreasing the file size, we offer a unique near-lossless compression approach in this study. We present a two-bitstream near-lossless compression pipeline that limits distortion in the task space at the burst level and in the image space at the frame level using the Lipschitz condition. Super resolution is a prominent downstream job in burst processing, and experiments on a real-world dataset show that the suggested technique reduces rate-distortion in both the burst frame space and this space.

Optimizing for l1 or l2 loss in RGB 444 format and measuring success using RGB PSNR are standard practises when it comes to assessing the efficacy of learnt image/video compression algorithms that preserve a high level of detail. It is generally known that optimising for a fidelity criteria leads to fuzzy pictures, which is often remedied by adding a content-based and/or adversarial loss terms. The problem with conditional generative models is that they lose accuracy. Here, we present a straightforward method for achieving this goal by proposing the use of gained variational auto-encoder (gained-VAE) in the luma-chroma (YCrCb 444) domain for learnt flexible-rate coding, which allows for sharper pictures to be obtained without any loss of fidelity. Our ability to observe that Y PSNR correlates with image sharpness better than RGB PSNR allows us to implement image-adaptive luma-chroma bit allocation during inference, i.e. to increase Y PSNR at the expense of slightly lower chroma PSNR to obtain sharper images without introducing colour artefacts. In addition, we point out that the suggested inference-time image- adaptive luma-chroma bit allocation technique may be implemented into any VAE-based image compression model. Compared to state-of-the-art models optimising RGB MSE at the same bpp, experimental findings reveal that sharper pictures with superior VMAF and Y PSNR may be produced by optimising models for YCrCb MSE using the suggested image-adaptive luma-chroma bit/quality allocation.

The need of being able to identify tampering in JPEG pictures has grown with the use of this compression format. Many data-driven methods are used to spot tampering in an uncompressed setting; however, the performance drops dramatically when photos are recompressed, which is a result of lossy compression. In order to do this, we create a deep residual framework that is capable of detecting tampering in re-compressed photos with the greatest quality possible (MDRNet). There are three stages that make up the framework: noise residual extraction, feature extraction, and classification. Initially, the front-end detector is greatly augmented by the noise residual extraction step, which adds three residual blocks with skip connections that may create noise residuals by reducing the picture content and amplifying the manipulation traces. The features of deep manipulation are then obtained using two efficient residual blocks and a cross feature learning approach, before being supplied to fully connected layers for classification. In order to improve forgery detection in real-world scenarios, experiments are conducted on 10 changes, each followed by a variety of quality variables. In addition, the proposed MDRNet outperforms state-of-the-art baselines in the toughest scenario of lossy post-JPEG compression.

Because of the vast quantity of data they contain, hyper spectral pictures need more memory for storage, processing, and transmission. One of the more efficient methods of compressing hyper spectral pictures is to describe them as a three-dimensional tensor. The term "tensor" refers to a structure with several dimensions. Many fields, such as numerical linear algebra, chemometrics, data mining, signal processing, statics, data mining, and machine learning, make use of tensors. Several tensor decomposition algorithms have been developed for dimensional reduction of tensors; they may be used to Hyper spectral image compression. The Hyper spectral Image is compressed using the Discrete Wavelet Transform and the Higher-Order Orthogonal Iterative Tucker Decomposition technique, as recommended. There were three more tensor decomposition methods compared to the simulation findings. Pavia University (610 x 340 x 103), Indian Pines (145 x 145 x 200), Salinas (512 x 217 x 224), and Abu-beach (all true hyper spectral pictures) were utilised in the experiments (150 X 150 X 102). After processing, the PSNR and SSIM are used to assess the Hyper spectral pictures' perceptual quality. Good PSNR and SSIM at high compression are offered by the suggested technique, which distinguishes it from the other three approaches.

Research on lossless picture compression has been fruitful for a long time. Several methods have been developed throughout time to approximate the smaller data set. In the past, techniques like the discrete wavelet transform (DWT) and the discrete cosine transform (DCT) were used to compress pictures, but now, many more options are available, including those based on machine learning and deep neural networks. In this study, we evaluate the effectiveness of many popular lossy image compression methods, such as Autoencoders, PrincIOTl Component Analysis (PCA), K-Means, and Discrete Wavelet Transform, using the Kodak Dataset (DWT). Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Compression Ratio (CR), and Structural Similarity Index are the metrics utilised to evaluate the suggested research (SSIM).

Lossy image compression is presently used extensively in many different applications. It's possible to provide a far higher compression ratio than with lossless compression, albeit distortions are produced ineluctably. These distortions have different characteristics depending on the coder used, the picture being compressed, and the compression settings. Visual perception and subsequent processing efficiency of compressed pictures are both impacted by distortions. As a result, it is important to have a firm grasp on the statistical and spatial spectral features of induced distortions in order to account for effects seen in compressed image processing and, maybe, enable their modelling. In this study, we outline the methods for analysing distortion qualities and provide an example using the AGU coder's compression of grayscale pictures. It is shown that the intensity of distortions is often higher in locally active regions, indicating that distortions have spatially variable statistics. Moreover, for all quantization steps and, accordingly, compression ratios evaluated, distortions are spatially uncorrelated. Tests are conducted with four representative examples of remote sensing imagery.

In this work, we introduce the 16-point discrete cosine transform (DCT) that eliminates the need for an orthogonal multiplier. While designing the suggested transform, we prioritised making it computationally efficient so that it may be used for efficient picture compression. With just 38 adds and no multipliers or bit shifts, the proposed quick transform technique is very efficient. Measures of computational complexity and image coding performance were used to evaluate the suggested transform. From what we can tell, the suggested transform has the greatest cost- benefit ratio of any state-of-the-art convert currently available in the literature. Proposed 16-point system's efficacy The PSNR and SSIM were used to assess the quality of the DCT approximation.

2021

The massive amounts of data that future PolSAR flights are predicted to acquire may considerably increase the cost of storing different geospatial cloud-driven applications. This cost might be mitigated by the use of data compression methods, such as those specified by the JPEG2000 (JP2) standard. Nevertheless, the performance of the target application must be monitored once these strategies have been implemented. In this study, we examine how JP2 and JPEG compression affect the classification performance of PolSAR data and find that neither format has a noticeable effect on the performance of Deep Neural Networks (DNNs).

As the Information Age has progressed, so has people's reliance on computers as a primary source and method of processing data. In today's digital world, sending many media files at once may be a real challenge for a computer system. Fast DCT-based JPEG picture compression is presented in this study. The algorithm explains how to encode and decode JPEG images. The image encoder can take a picture in BMP format, run it via JEPG, and produce a binary file suitable for real-time storage. With the proper decoding software, the picture may be recovered from its compressed state. Moreover, the JPEG format can be used to encode a static image during image transmission, capitalising on the fact that human vision is insensitive to chroma, and converting the original JPEG image's colour RBG into brightness y, chroma Cr, and CB to achieve compression while also effectively decreasing the amount of chroma data.

The proliferation of smartphones and other portable electronic gadgets that serve as convenient methods of communication and entertainment is a clear indication of this trend. In light of the fact that the vast majority of currently available smartphones include an integrated biometric identification mechanism, it is imperative that these gadgets be protected. There is a wide variety of facial characteristics, and many of them are shared with other individuals, particularly close family members. The face authentication system and the recognition photos must thus be sent to the cloud for processing. In order to facilitate the implementation of a quick and effective facial image compression technique, which will be discussed in detail in the forthcoming publications, this research article has analysed the prior works based on the facial image compression paradigm for their improvements and limitations.

Nowadays, online video makes up the vast bulk of the data sent over the internet. As the data rate needed to store uncompressed visual data is so high, lossy compression methods are used to reduce it. Cloud-based analytics (classification, detection, etc.) are increasingly relying on the ever-growing volume of visual data being produced. Visual distortions created by image and video compression, particularly at lower data-rates, may significantly reduce performance on such analytical tasks. Moreover, many picture and video compression methods allocate additional bits to elements that are seen as more important by the human eye. These qualities, however, may not be the best option for semantic tasks. Here, we offer a method for compressing visual data in a way that improves performance on a specific analytical job. To discover the appropriate embeddings, we train a deep auto-encoder with a multi-task loss. For better accuracy in inference, we employ an approximation differentiable model of the quantizer during training. We demonstrate that our method outperforms JPEG compression when applied to an image classification issue, at a certain degree of compression. Furthermore, compared to the state-of-the-art method in this area, our method significantly exceeds the competition.

This study proposes a system for point cloud geometry and attribute lossless and lossy coding that is based on a sphere projection. In order to simulate the local geometric structure of the original point cloud, we adaptively partition it into blocks and then generate a sphere that is a good match for each block. To convert a 3D point cloud to a collection of range pictures, we present the sphere coordination transform and the spherical projection technique. To further reduce the resulting range pictures, a new compact representation based on Morton codes is developed, which splits the images into occupancy images and attributes vectors. The experimental findings show that the suggested technique outperforms geometry-based point cloud compression for lossless compression of LiDAR point cloud datasets (G-PCC). The suggested technique outperforms Draco in terms of rate-distortion (R-D) for the object point clouds datasets while using lossy compression.

In order to reduce the size of the data associated with a point cloud, current methods use geometric or video-based compression algorithms. We investigate a completely new strategy prompted by the most recent achievements in learning to represent point clouds. The 3D point cloud may be seen as a 2D manifold. To be more precise, we use an unique optimised mapping approach to fold a 2D grid onto a point cloud and then transfer characteristics from the point cloud onto the folded 2D grid. This mapping yields a picture, which allows for the use of conventional image processing methods on point cloud characteristics. As this mapping is inherently lossy, we offer a number of methods for improving it such that characteristics may be transferred to the 2D grid with as little disruption as possible. In addition, this method may be used in a versatile manner to point cloud patches to better adjust to regional geometric complexity. Here, we focus on point cloud attribute reduction, therefore we use a standard 2D image encoder to reduce the size of the resulting picture. First experiments with the suggested

folding-based coding technique indicate promising results, with results competitive with the most recent MPEG Geometry-based PCC (G- PCC) codec.
Comprehending the unique and difficult requirements of point cloud compression is no easy task. Existing video codec technology can be used by video-based systems for effective compression, but only if the cloud's 3D geometry and characteristics are provided in a suitable, regular 2D grid. We provide a technique for making square video blocks of uniform size that may be used to extract colour information from point clouds. Traditional video codecs may make effective use of colour compression when applied to a picture made up of our video blocks. The main emphasis is on creating a computationally efficient voxel-to-image projection technique. Comparison against MPEG's G-PCC reveals comparable performance, with average PSNR increases of 0.68dB and attribute (colour) bit rate reductions of -15.10%. (TMC13 v5.1).

In this paper, we shall analyze different features of compression using machine learning and DL algorithms with IOT and IHT structures. The proposed design with IOT and IHT is implemented with Intuitive Sigmoid Predictive Transform. For image compression, we have proposed two algorithms with a sigmoid prediction feature and Intuitive Filter Semantic Masking for residual error loss and compression ratio. An IOT algorithm with a dense layer architecture and a sigmoid predictive function is implemented on autoencoders to compress the images and image frames accordingly. Overall, three algorithms are proposed with layered features that estimates conditional probabilities on CNN, AE, and Dense-DL.

3. Image Compression

3.1 Concept

Cloud Storage is gaining more attraction as a part of cloud computing paradigm as it enables user to access their data whenever and wherever needed, once it is stored on cloud. Moreover, cloud provide huge storage which attracts corporate, business data as well as medical data to be stored on cloud. With the advancement of information and computing technology, large scale datasets are being generated today. The example datasets are medical images, remote sensing images, satellite image database, etc. These are needed to be handled properly due to its privacy- sensitive nature. Thus, it is of critical importance that security should be embedded in this image services. With these feature analytics, the researchers in [4-20] have depicted multiple algorithms to realize the importance features of image compression indicating the different approaches.

3.2 Machine Learning algorithms

3.2.1 KMEANS

1. Concept

In a colored image, each pixel is of size 3 bytes (RGB), where each color can have intensity values from 0 to 255. Following combinatorics, the total number of colors which can be represented is 256*256*256 (equal to 16,777,216). Practically, we can visualize only a few colors in an image very less than the above number. So, the k- Means Clustering algorithm takes advantage of the visual perception of the human eye and uses few colors to represent the image. Colors having different values of intensity that are RGB values seem the same to the human eye. The K-Means algorithm takes this advantage and clubs similar looking colors (which are close together in a cluster)

2. Algorithm

The overall algorithm discrIHTion for the K means is decribed below:

Algortihm 1: K-means

a. Speciify the number of the clusters and denote as K
b. Randomly assifn centroid values for K clusters
c. Repeat:
    a. Expectation: Assigning eacj point to closest centroid value:
        i. $F(k \in K(i)) = P(i)$
    b. Maximization: Compute the overall new Centroid Mean from F closer to the k_i
d. Until: The change of values of the centroid is consistant

3. Implementation

a) Image Input: Load the image from the disk.
b) Reshape Input Image: The size of the input image is (rows, cols, 3), flatten all the pixel values to a single dimension of size (rows*cols) and the dimension of each pixel is 3 representing RGB values. The size of the flatten image will be (rows*cols, 3).
c) Clustering: Implement the k-Means clustering algorithm to find k-centroid points that represent its surrounding colour combination.
d) Replace each pixel with its centroid points:All the colour combination of (rows*cols) number of pixels is now represented by its centroid points. Replace the value of each of the pixels with its centroid point.
e) Reshape Compressed Image: Reshape the compressed image of (rows*cols, 3) dimensions to original (rows, cols, 3) dimensions.
f) Output Compressed Image:Display the output image and store it to disk.

Even though the overall design model with K-means clustering features with image compression have indicating the overall importance of how K is designed and randomly chosen depending upon the values of the clustering of the image features for compression. Though these are effective with simpler aspects of the applicating requirements indicating the overall changes in the design. Hence, proven less effective for high quality images. To encapsulate such feature hybrid algorithms have been introduced with DWT and DCT indicating the compression feature with depth compression and quantized error models.

3.1 DCT-DWT

3.1.1 Concept

The overall perspective of the DCT-DWT algorithm is to provide two stage compression model with DCT and DWT for better security features of the image in clouds or even in servers. The process of the transformation on the image is represented with flow diagram below indicating the working features on the images with its formulations.

3.2 Flow diagram

In figure 1, representing the overall structural flow model of the DCT-DWT hybrid design of the image compression method. In this design, the authors in [3-7] have depicted with different pre-processing filters and techniques for

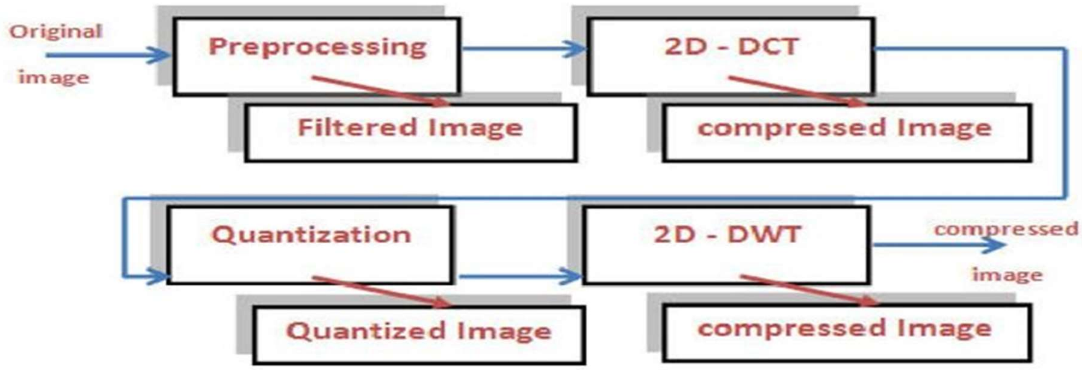

Fig.1 Representing the overall DCT+DWT hybrid block diagram for Image compression

3.3 DEEP LEARNING ALGORITHMS

3.3.1 Auto Encoders

It is possible to train neural networks to compress and encode data and reconstruct the received data with minimal possible error changes. A Variable Auto Encoder structure for image reconstruction is illustrated in Figure 4. The Autoencoders are neural networks that are precise and feasible solution. The representation of encoded data with an autoencoder implicates the possibility of learning the optimal higher dimensional space in an unsupervised way.

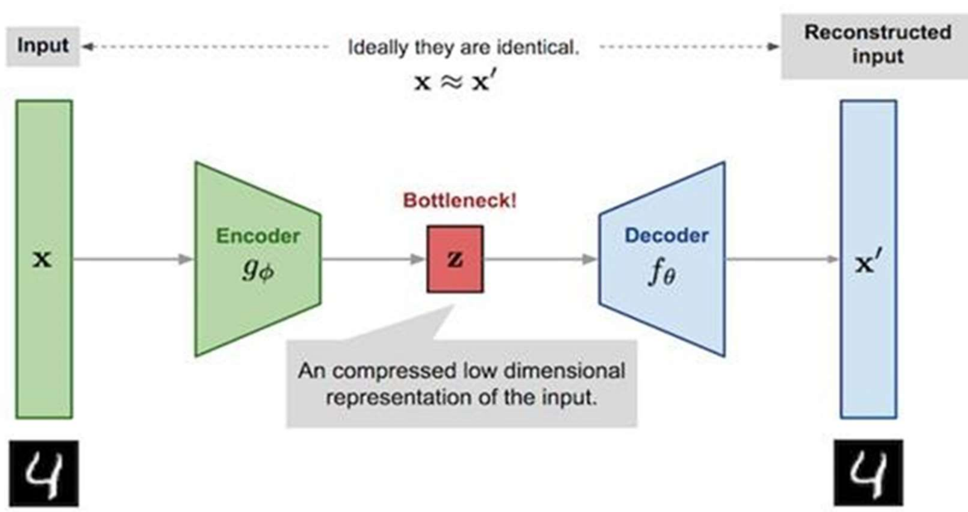

Fig. 2 Representing the Variable Auto Encoder structure for image reconstruction.

The Auto Encoder structure comprises two components:

1. Encoder-Decoder.
2. Bottleneck (Code Layer).

Since the structure for efficient compression from the input phase to the output phase would reconstruct the different weights for encoder and decoder with bottleneck layer optimization, one such feature for this layer is estimated with the below formulations:

$$z = g_\theta(x) \qquad (1)$$

The reconstructed input is written as follows:

$$x' = f_\varphi(g_\theta(x)) \qquad (2)$$

The overall reconstruction loss calculated using mean square error is given by eq (3):

$$L_{AE}(\theta, \varphi) = \frac{1}{n} \sum_{i=0}^{N} (x_i - x'_i)^2 \qquad (3)$$

Table1 The Summary of selected state-of-the-art algorithms

Sr. No.	References	Algorithm	Pros	Cons	Research gap
1	Zhang et.al [1]	Residual Learning	An intuitive algorithm remove Gaussian noise observed af tercompression	There is no denoising feature with Gaussian.	For any unusual cases, noises with random models on Rayleigh and passion distribution must be implemented.
2	S. Perry *etal*[2], R. Birkeland et.al [5]	Semantic-oriented learning	Image compression semantic compression model with detailed architecture	Encoder and decoder structures are complex with more optimization of L1 and L2 norms.	Video compression-based semantic models have to be developed with intuitive memory optimization algorithms.
3	O. U. Ulaset.al [13], Y. Heet.al [23], Papadopoulos et.al [24]	Switchable Texture-Based Video Coding	Multi scale d architecture w ithMEMC	More traffic usage for memory in DL to analyse the solution of compression, either lossy or lossless.	QoE metric model generalization N PU neural processing unit
4	Busson et. al[15], Chen et.al [16]	Deep Learning-Based Prediction Models	This method uses a decoder to store and stream videos in low quality and display them in high quality without changing the MPEG compression algorithm, resulting in less disc space and bandwidth usage.	It fails to achieve better results when compared with VVC and HVEC with different performance factors.	To apply current trends coming from the DL field (e.g., spatial and channel-wise attention, and transfor merencoding) to extend the model. To test whether these techniques are useful for the DCT coefficients restoration task.
5	Habibian et.al[17],	Rate-Distortion	To bridge the gapbetween	None	None

Song et.al [22], Theis et.al [18]	Autoencoders	semantic video understanding and compression by learning to allocate more bits to objects from categories of interest.	

Table 1 depicts the overall comparison and its importance with the memory optimization feature on DL to be based on the state of arts (SOA).

In light of the above literature, it can be said that the overall compression efficiency matters for different resolutions. Hence, a predictive approach with other algorithms must be estimated with this efficiency and analyzed with the data accordingly.

4. INTERPOLATIVE DEEP INTUITIVE NETWORK

4.1 Concept

As analysedwith state of art techniques, the design and its implementations on the image compression models have been realized with in this IDIN model indicating the improvement of the image compression feature in cloud systems. Presently, to realize such feature we have opted local cloud indicating the overall samples sizes about more than 2 GB files with varied resolutions. Our proposed work, postulates two transform techniques indicating the overall adverse loss effect while compression via interpolative approach. The transform of IHT indicates the importance of the interpolative features on the pixel values to reduce the overall dimensionality with decomposition blocks sizes of $K(f, g)$ as the dimensionality matrix after decomposition. This outcome indicates the realized features of the pixel's differences on the original images with diagonal orthogonal matrix decomposition via IOT technique which generates $Z(f, g)$ with orthogonality of diagonal matrix operations. All these observed values from IOT approach are then applied to the proposed AE inculcating the compression model with Interpolative deep dense model. This model comprises of encoder and decoder as the building block for compression and decompression feature. While the code block in proposed work, implicates how better filtered data outcomes are observed as the post processing.

4.2 Design Process

a) To improvise a filter approach for Pre-processing the image data

b) Indicate the overall filter loss via segmentation approach

c) Implore the interpolation feature on the input images for dimensionality reduction via IHT technique.

d) Implement a Transform for the reduced dimensionality model based on diagonal orthogonality principle.

e) Effective AE network with IDDM model is implemented to indicate the best compression with least loss.

f) Finally, tabulate with State of art techniques.

4.2.1 Dataset

The 1000 image samples are utilized with real-time as photographed from i-phone-14, also the images from CIFAR10K and MNIST datasets are used to test the effectiveness of the proposed method. We test the efficacy of the suggested strategy by analysing the MNIST dataset, which comprises pictures of numbers and characters for recognition characteristics, using IOT in conjunction with many other DL techniques. For best results when using CNN, Dense DL, and AE, it is recommended to use the suggested models that include IOT and IHT. To guarantee the distinguishable shifts in DL algorithms, the formulation for each layer is with IOT weights.

4.2.2 Block Diagram

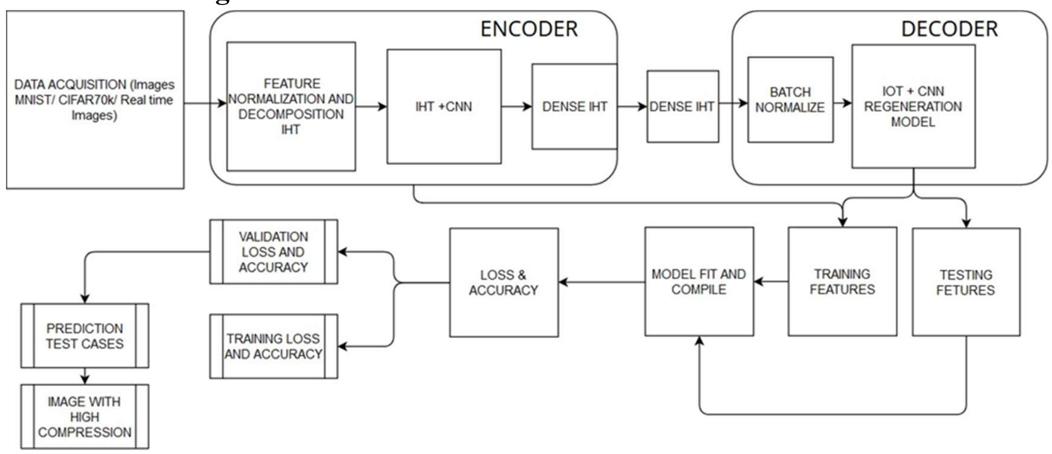

Fig. 3 Representing the Overallproposed block diagram for Encoder-Decoder Model for IOHPT-AE.

4.2.3 Working Approach

Figure 6 depicts the general architecture of the suggested model with encoder. As shown in equation 10, the sigmoid filter is used to clean up the three channels of the input pictures. The filtered output is then sent to an 8-layer architecture that uses convolutional layers with IHT-IOT, dense layers with IHT-IOT, and ReLU with batch normalizations layers, all of which have distinct potential connections. It's clear what has to change when going from IOT to IHT, and vice versa; otherwise, you can't transform a 1-dimensional vector. To declare the IOT as error functionality in 2-D conv, we have used a similar method to the one used to define the functionality of the 2-D conv layer, whose job it is to convert time scale to frequency scale. Using the observed weights on each RGB value for each picture and video frame, the IOT conducts compression. The state-of-the-art models, such as GAN, LSTM, BILSTM, etc., are compared to real-time pictures using Kaggle Website-based datasets. As can be seen in Figure 3, the resulting tangled image is 2-dimensional (displaying just grayscale), but the overall compression characteristic is enhanced thanks to IOT, as will be shown later.

For its convolutional processing, the Conv-2d layer accepts images with input dimensions of 1280x720, 1024x480, 640x480, and 1920x1080. Convolve stands for "changing domain," which may be either "high frequency component" or "low frequency component." Convolutional HC and LC traits are passed on to IOT or IHT. In the case of IOT, the HC or LC characteristics are determined by using the remainder theorem, which assigns certain values of remainders to the reconstruction of HC or LC. The picture is processed using the IOT-IHT function, and the three-layer architecture of the Encoder shows how integral connection is to the design. Although IOT and IHT tend to increase pixel-level compression, we apply convolution with a filter order of 8,16,32. Throughout the Encoder and decoder's training and testing phases, the general structure of filter values is represented.

Similar to how 2D data may be compressed to 1D data using thick layers in the coding block. Although encryption is best accomplished with a thick layer, the ReLU is used to reduce the exponential rise in computation using a linear method. Data inside the network structure is normalized using the batch normalization layer.

4.2.4 Decoder

In Figure 3, we see a block diagram depicting the Decoder's implementation of a 2-layer model including CNN dense layers. In order to instantiate the various embedded places and clusters, the decoder portion of the image/frame is encoded with dense layers and normalized. Depending on the complexity and estimate loss determined using IOT, this facilitates the generation of entirely lossy/lossless pictures. To estimate the global loss function, we use eqn(12), a variant of the swish function provided by eqn(11) (11). Convolution's primary purpose, as seen in Figures 6 and 7, is to alter the Image's domain, whereas in the decoder, we first do inverse convolution, also known as "de-convolve," to transform the frequency domain into the time domain. An 8-layer decoder architecture is shown, with a progression in filter size from 8 to 16 to 32 to show how 2- and 3-dimensional results are obtained from 1-dimensional input vectors using inverse convolving. Starting with the Loss functions determined in Section 2, we create a custom layer architecture that details the various convolution and inverse convolution layers on E-D with IOT-IHT. Figure 13 depicts one of the two outputs produced by the AE decoding process at the beginning of the process. In order to improve the dense layer and its relative loss characteristics, the IOT method proposes assigning the feature weights to the activation layers of the custom model. Typical results from the Encoder-Decoder system need the development of loss functionality at each layer in order to analyses the model's potential for improvement. To begin, the Inverse Convolution Block uses the Loss Characteristic Formulation from IOT-2 to assess whether the Encoded Image should be converted to 2D Convoluted Data. We give a simple cutoff number that holds true throughout all time periods studied, which range from the present to the distant past. When applied to 1-dimensional RGB data, these thresholds determine the predicted loss for each area, ensuring that the proper colors are shown. With a loss of no more than one percent every iteration of the epoch, we implement the transformation from IOT to IHT. Because of the double transform capability, the design metrics may have their R, G, and B components converted independently before being merged during normalizations. Lastly, the Decoder layers' activation feature may be used to highlight the overall classification if necessary, and the compression normalization feature reduces the number of training epochs for a quicker result that allows the layers to function autonomously.

4.3 Algorithms

4.3.1 IOT Transform

Algorithm 1: Interpolative Orthogonal Transform

$Inputs: Images\ (X_I, Y_I\ (predicted\_img))$

$Output: Loss\ Interpolative\ Estimation$

While $E\{(Log(X_I), Log(Y_I)\} < \varepsilon - x_i \;||\; E\{(Log(X_I), Log(Y_I)\} < \varepsilon - y_i\ do$

$loss_{out} \leftarrow 0$

$while$ $(\log(X_I) > \sigma_{minx}, \log(Y_I) < \sigma_{miny})$

$X_i \leftarrow y_i * X_i$ (From 11)

$E(P(x_i) = X_i + \mu * W_{i+1}$ (μ is weight factor) (From 14)

$E(loss_{mx}) = \int_{-\infty}^{\infty} X_I * E(P(x_i))dx$

$E(loss_{my}) = \int_{-\infty}^{\infty} Y_I * E(P(y_i))dy$

$loss_{out}(i) = \sum_{i=1}^{N} (E(loss_{mx} * loss_{my})^2 - E(loss^2_{mx})\,E(loss^2_{my}))$ (26)

End

End

4.3.2 IHT TRANSFORM

Algorithm 2: Interpolative Heuristic Transform.

4.4 Formulations

We are considering the image $\{x(i, j)\}$ as the input images where i, j represents the row and column of the image dataset. Here {[]} represents the set of images for the given set of iterations considered. The proposed transformation analysis for the images, considers the linear model of 3D space for color space on red, green, and blue image models. The overall prediction weights with higher dimension are considered for Images. With the ideal solutions of the Image compression and its design characteristics, we have proposed an IHT model on the current set of regions of space color models using the 3D and 2D scenarios. The desired color space equations and their prediction feature for frames as the linear model are given by eqn(16).

$$y(i,j) = \sum_{i=1}^{M}((\sum_{j=1}^{M} W_{i,j}^{T} x(i,j)) + v(i,j)) \tag{4}$$

Here $y(i, j)$ represents the prediction model for each image, W is the weight formulation on a set of the image, v is the generated predicted pixel values, and T represents the transpose of the matrix.

$$W_{i,j} = [w_{10}, w_{02}, w_{30} \ldots\ldots\ldots w_{L-1}]^1 * [w_{01}, w_{02}, w_{03} \ldots\ldots\ldots w_{K-1}]^1 \tag{5}$$

Equation (5) presents the weight vector of the neural network model and has a length of L and K. The $x(i, j)$ is an input frame observed from the input video in use. The estimated pixels values are obtained from the prediction algorithm as the Bayes model given by:

$$P\left(\frac{x}{v}\right) = p(x \cap v) * \frac{p(x)}{p(v)} \tag{6}$$

The probability of at least one prediction for the given frame is true $p(x \cap v)$, and for each set of predictions
would be in input, the x would be $P\ (x)$. The random model with a finite set of A and C is the solution for each v predicted value of x and v random variables. To yield the prediction on v $(Random\ variable)$ on each pixel of x, we have

$$P(v) = E\{\mu * \sum_{k=1}^{M} (x(k,j))_k + \delta * \sum_{j=1}^{L} x(k,j)_j\} \tag{7}$$

The parameters $\mu\ and\ \delta$are the step size of the linear prediction model in 2D for the conventional image (BW or RGB). Here, A and C are the values and weights observed from equation (7) when expanded in exponential form as:

$$E\{P(v)\} = E(H(i,j) + G(i,j)) => E\{P(H)\} + E\{P(G)\} \tag{8}$$

The $P(x \cap v)$ is the intersection value for the prediction and input values where each value observed is estimated with V=C. Hence C is the solution value observed from equation (20). Hence, the resultant equation would be

$$E\{P\ (\frac{x}{v})\} = E\{p(x \cap v)\} * \frac{E\{p(x)}{E\{p(v)\}} \tag{9}$$

$$W_{i,j} = \alpha * (\frac{E\{p(A)}{E\{p(C)\}}) \tag{10}$$

$$v(i,j) = W_{i,j} * E\{P(v)\} \tag{11}$$

From equations (4) and (7-11) we have

$$y(i,j) = \sum_{i=1}^{M} ((\sum_{j=1} \{\mu * (\frac{E\{p(A)}{E\{p(C)\}})\} x(i,j)) + \delta * (\frac{E\{p(A)}{E\{p(C)\}}) * E\{P(v)\}\}) \tag{12}$$

The terms in equation (12) depicts the α as contrast and δ as intensity values varying from (0,1) for each type of image or frame chosen.

5. RESULTS AND DISCUSSION

5.1 MNIST Dataset Results

5.1.1 Existing Autoencoder

Image compression and reconstruction with Autoencoder have been utilized in many different features of input images that were considered with estimated weights. This structural design with the generic AE has utilised a specific layer features to create the image compression feature using encoding and decoding of the images. In MNIST, 60K images are gathered with different labels from 0-9, and a reconstruction feature with output images are declared by estimating different weight equations as mentioned in equations (10-14). From Figure 8, we can observe that the simple AE based approach would not provide the correct labels with the reconstructed image.

Fig.4 Samples of MNIST dataset a) Original image b) Decompressed image

5.2 Proposed IHPT-DNN

This AE implements both the functionality of IHPT on the different feature reconstruction models by transforming different images, and its encoding and decoding models as stated in section 3. The proposed model implements the reconstruction feature based on two different datasets, which are CIFAR-10 and MNIST. The image compression using the proposed AE utilizes three steps of the design model:

At the dataset declaration stage, our design feature is to process the different images as described based on the label type and reconstruct the image accordingly. We instantiate the design on the training features with a transformed formulation as per the algorithm stated. We have used Algorithm-1 to remove noise and other distortions of the images. The original feature of the images with different class features from 0–9 is represented in fig.9.In the training model, we implicate the other features of the images by utilising the encoder and decoder structure for the model that must be designed.

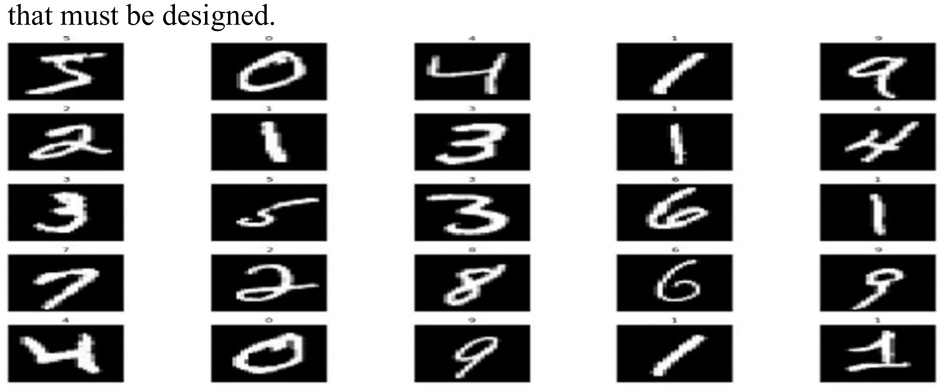

Fig.5 Representing the original Image for the MNIST database.

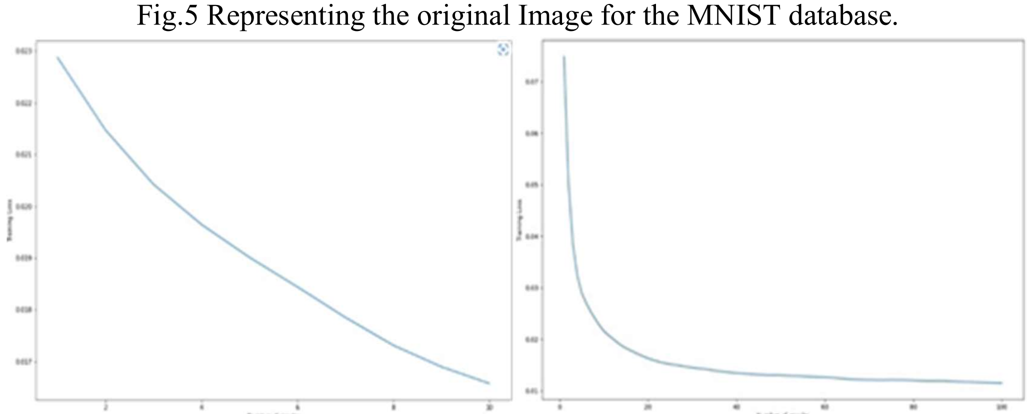

Fig.6 Representing the Training Loss curve with 10 and 100 Epochs

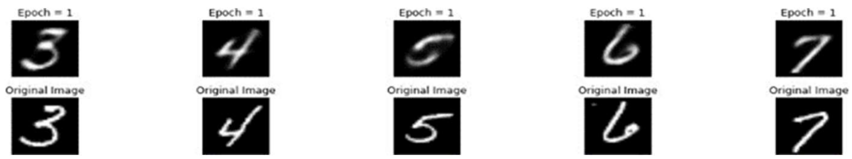

Fig.7 Reconstructed image with 1 Epoch

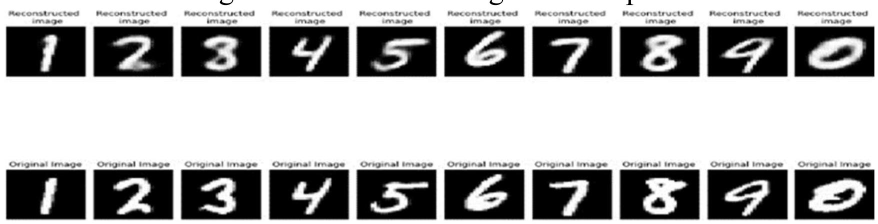

Fig. 8 Reconstructed image with 10th epoch

Fig. 9 Reconstructed Image with 100th Epoch

In the testing phase of our proposed IOT-IHT algorithms, different image features are reconstructed with different levels of segmentation and their feature modelling with noise removal filters. The sigmoid function with filter weights are improvised on the different epochs ranging from 1–10 and 1-100 for more precise image reconstruction via the proposed AE. For the initial epoch, we could observe in Figure 9, the blurred results of the input samples and their feature illustration. While in epoch five, we could see the correct response of the test dataset images with the noisy model. As the epoch increases, the image reconstruction with 100 epochs has similar results and outstanding SSIM values as mentioned in Table 2.

5.3 CIFAR 10 Results

The CIFAR-10 dataset (CIFAR-10 and CIFAR-100 datasets (toronto.edu)) [26] has 60000 images. The proposed AE has been used for image compression on CIFAR-10 dataset to encapsulate different features of the images, attaining minimum loss as mentioned in Table 2. The overall estimated loss is observed with different testing ranges of inputs chosen during classification. So, the minimum overall loss is observed when 20% of images are used for testing and 80% for training.

Table 1 MNIST database test loss.

SNO	*Image (% Test)*	*Algorithm (Testing)*	*Loss*
1	10%	IHOPT-AE	0.03841
2	20%	IHOPT-AE	0.01737
3	30%	IHOPT-AE	0.01984

5.4 Training and

5.5 Testing Phase

In the training phase, collective features for training and testing datasets are utilized with the generated model. The layer implementation and its representation for the design are designated above. The above featured values of the layers are indicating the 8,16,32-bit compression feature based on the E-D structure in figure 6-7. Our design on CIFAR database utilized 32X32 compression filter as the convolution model indicating less reconstruction loss at 1%. We have tested with 10000 images other real time images with high resolutions depicting the overall pixelated values with 15x15 or 32x32.

The training and testing phases comprise on the loss estimated values for each compression feature. This loss values are formulated in section 2.4.

Fig. 10 Representing the Reconstruction of Images for CIFAR10 with compression Image with 15x15 with 12.5KB in BMP

In the training phase, the design is estimated with different DL algorithms with its model created from the tensor flow library. The model created with the Tensor flow would emphasize the estimated parametric weights and its validation accuracy for compression of data with images.

Table2 Representing the Training and Testing Loss for IHOPT-CNN Model

Iterations	*Training loss*	*Testing loss*
1500	0.4657	0.5321
3500	0.3452	0.4532
4500	0.2115	0.3780
6000	0.1985	0.2156

Table3 Deep-dense layer loss estimation using IHT

Iterations	*Training loss*	*Testing loss*
1500	0.4245	0.5834
3500	0.119	0.2245
4500	0.0185	0.0319
6000	0.00462	0.0032

Table 4 Representing the CIFAR dataset Training and testing phase.

No of Epochs (Testing)	*Epoch avg. loss (Testing)*	*Accuracy (Testing)*	*No of Epochs (Training)*	*Avg. loss (Training)*	*Accuracy (Training)*
2	0.01528781	99.98471219	1	0.014813	99.985187
4	0.01151757	99.98848243	2	0.014083	99.985917
6	0.00937661	99.99062339	3	0.015076	99.984924
8	0.00946861	99.99053139	4	0.014269	99.985731
10	0.00891677	99.99108323	5	0.016616	99.983384
12	0.00913224	99.99086776	6	0.037616	99.962384
14	0.00884325	99.99115675	7	0.016883	99.983117
16	0.00861413	99.99138587	8	0.017671	99.982329
18	0.00857244	99.99142756	9	0.027874	99.972126

Fig. 11 Representing the Overall 3 Images from Flicker 8k dataset images are chosen for Compression.

In Figure 11, we have observed a 32-bit frame representing the eight epochs with each set of pixel reconstruction using IHT and IOT algorithms. Since the epochs are less, their iterations with the proposed model have resulted in different possible 32-bit frames of output. At the same time, the complete reconstruction of the image with 1280x720 and 720x720 is reconstructed with every single image. These reconstructed images are clustered with the conditional features on the epoch loss with a range from 0.00009 to 0.001 is shown in Table 4. The resultant values for the design with different bits per pixel are utilized with the number of bits used to create the compression feature. Hence, the epochs from the range of 2–18 are presented with the average loss. The overall accuracy is calculated based on the formula as mentioned below.

$$Accuracy + loss = 100\,\%\ design\ outcome$$

The training of the AE, DL with IHT-IOT feature represents the overall least error observed which is less than 1% as shown in table-4. For testing twice of the epochs are needed to optimize the loss at reconstructed outputs as shown in figure 11.

6. CONCLUSION

This paper addresses the need for memory optimization problems effecting the layer errors and their occupancy while iterated with a greater number of consecutive operations. So, addressing this issue, we postulate a novel feature with IOT and IHT estimation features for weight prediction on the deep learning structures for compression of images and video frames. Since memory design and its improvements have been more frequently addressed and have raised the barrier of cost predominantly. Indicating such complexities in memory addressing and its involvement with data filtering is a difficult task. So, to resolve the complexity and add more functionality, an AE- IOT model is implicated for the compression feature. The functionality of the design is addressed with the types of algorithms utilized to reduce the output compression error and internally for each layer. This feature is depicted with the proposed AE-IOT encoder-decoder architecture shown in section 3. The proposed encoder and decoder models are used for reconstruction of images and video frames to estimate the coding layer model. The effective featured weights are demonstrated with equations (4) and (11) which have enhanced the Deep Learning architectures formulated with IOT and IOT algorithms. The proposed network model with IOT-IHT ensures the learning feature with a rate of 10^(-2) and 10^(-3) for IOT-CNN, IHT-DL model while IOT-AE have implemented with factor 10^(- 7)as tabulated in tables 2-4. The experimental results tabulated with the performance characteristics are implicated to emphasize the effectiveness.

REFERENCES

1. M. C. H. Zerva, V. Christou, N. Giannakeas, A. T. Tzallas and L. P. Kondi, "An Improved Medical Image Compression Method Based on Wavelet Difference Reduction," in IEEE Access, vol. 11, pp. 18026-18037, 2023, doi: 10.1109/ACCESS.2023.3246948.

2. S. Perry et al., "Subjective and Objective Testing in Support of the JPEG Pleno Point Cloud Compression Activity," 2022 10th European Workshop on Visual Information Processing (EUVIP), Lisbon, Portugal, 2022, pp. 1-6, doi: 10.1109/EUVIP53989.2022.9922803.

3. D. Li, J. Wang and G. Li, "Near-lossless Point Cloud Geometry Compression Based on Adaptive Residual Compensation," 2022 IEEE International Conference on Visual Communications and Image Processing (VCIP), Suzhou, China, 2022, pp. 1-5, doi: 10.1109/VCIP56404.2022.10008796.

4. S. Wang, J. Jiao, P. Cai and L. Wang, "R-PCC: A Baseline for Range Image-based Point Cloud Compression," 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 2022, pp. 10055-10061, doi: 10.1109/ICRA46639.2022.9811880.
5. R. Birkeland, S. Berg, M. Orlandic and J. L. Garrett, "On-Board Characterization Of Hyperspectral Image Exposure And Cloud Coverage By Compression Ratio," 2022 12th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Rome, Italy, 2022, pp. 1-5, doi: 10.1109/WHISPERS56178.2022.9955117.
6. H. Chandra and S. Bajpai, "Listless Block Cube Tree Coding for Low Resource Hyperspectral Image Compression Sensors," 2022 5th International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT), Aligarh, India, 2022, pp. 1-5, doi: 10.1109/IMPACT55510.2022.10029076.
7. D. Kucherov, V. Tkachenko, N. Khalimon and O. Poshyvailo, "Signals and Image Processing in Information Systems by Tensor Analysis Methods," 2022 IEEE 4th International Conference on Advanced Trends in Information Theory (ATIT), Kyiv, Ukraine, 2022, pp. 277-280, doi: 10.1109/ATIT58178.2022.10024221.
8. A. Moon, J. Chen, S. W. Son and M. Kim, "Characterization of Transform-Based Lossy Compression for HPC Datasets," 2022 IEEE/ACM 8th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD), Dallas, TX, USA, 2022, pp. 56-62, doi: 10.1109/DRBSD56682.2022.00013.
9. M. H. Faykus, B. Selee, J. C. Calhoun and M. C. Smith, "Lossy Compression to Reduce Latency of Local Image Transfer for Autonomous Off-Road Perception Systems," 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 3146-3152, doi: 10.1109/BigData55660.2022.10020267.
10. A. Tawfik et al., "A Generic Real Time Autoencoder-Based Lossy Image Compression," 2022 5th International Conference on Communications, Signal Processing, and their Applications (ICCSPA), Cairo, Egypt, 2022, pp. 1-6, doi: 10.1109/ICCSPA55860.2022.10019047.
11. A. Mahmoud et al., "An Efficient Hardware Accelerator For Lossless Data Compression," 2022 5th International Conference on Communications, Signal Processing, and their Applications (ICCSPA), Cairo, Egypt, 2022, pp. 1-4, doi: 10.1109/ICCSPA55860.2022.10019048.
12. W. Jiang et al., "Task-Oriented Near-Lossless Burst Compression," 2022 IEEE International Symposium on Multimedia (ISM), Italy, 2022, pp. 257-260, doi: 10.1109/ISM55400.2022.00057.
13. O. U. Ulas and A. M. Tekalp, "Flexible luma-chroma bit allocation in learned image compression for high- fidelity sharper images," 2022 Picture Coding Symposium (PCS), San Jose, CA, USA, 2022, pp. 31-35, doi: 10.1109/PCS56426.2022.10017994.
14. V. K. Kadha, P. Deshmukh, K. C. Rayasam and S. Kumar Das, "Robust Manipulation Detection Scheme for Post-JPEG Compressed Images using CNN," 2022 IEEE 19th India Council International Conference (INDICON), Kochi, India, 2022, pp. 1-6, doi: 10.1109/INDICON56171.2022.10040157.

15. B. Sucharitha and K. Anitha Sheela, "Hyper Spectral Image compression using Higher Order Orthogonal Iteration Tucker decomposition," 2022 IEEE 19th India Council International Conference (INDICON), Kochi, India, 2022, pp. 1-7, doi: 10.1109/INDICON56171.2022.10040093.

16. A. Thakker, N. Namboodiri, R. Mody, R. Tasgaonkar and M. Kambli, "Lossy Image Compression-A Comparison Between Wavelet Transform, Princ IOTl Component Analysis, K-Means and Autoencoders," 2022 5th International Conference on Advances in Science and Technology (ICAST), Mumbai, India, 2022, pp. 569-576, doi: 10.1109/ICAST55766.2022.10039613.

17. V. Abramova, V. Lukin, S. Abramov, K. Abramov and E. Bataeva, "Analysis of Statistical and Spatial Spectral Characteristics of Distortions in Lossy Image Compression," 2022 IEEE 2nd Ukrainian Microwave Week (UkrMW), Ukraine, 2022, pp. 644-649, doi: 10.1109/UkrMW58013.2022.10036949.

18. V. P. S. Thoudam, T. Sera, X. C. X. Marak, M. Saddam and R. Lalparmawii, "Low complexity 16-point DCT approximation for image compression," 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2022, pp. 545-550, doi: 10.1109/ICCCIS56430.2022.10037721.

19. J. Checker, S. De, V. Turkar and G. Singh, "Performance Impact of JP2 Compression on Semantic Segmentation of PolSAR Images," 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 2021, pp. 339-342, doi: 10.1109/IGARSS47720.2021.9554881.

20. P. Dahiwal and A. Kulkarni, "An Analytical Survey on Image Compression," 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, 2020, pp. 656-661, doi: 10.1109/WorldS450073.2020.9210364.

21. W. Xiao, N. Wan, A. Hong and X. Chen, "A Fast JPEG Image Compression Algorithm Based on DCT," 2020 IEEE International Conference on Smart Cloud (Smart Cloud), Washington, DC, USA, 2020, pp. 106-110, doi: 10.1109/SmartCloud49737.2020.00028.

22. N. Patwa, N. Ahuja, S. Somayazulu, O. Tickoo, S. Varadarajan and S. Koolagudi, "Semantic-Preserving Image Compression," 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 1281-1285, doi: 10.1109/ICIP40778.2020.9191247.

23. Y. He, G. Li, Y. Shao, J. Wang, Y. Chen and S. Liu, "A point cloud compression framework via spherical projection," 2020 IEEE International Conference on Visual Communications and Image Processing (VCIP), Macau, China, 2020, pp. 62-65, doi: 10.1109/VCIP49819.2020.9301809.

24. M. Quach, G. Valenzise and F. Dufaux, "Folding-Based Compression Of Point Cloud Attributes," 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 3309-3313, doi: 10.1109/ICIP40778.2020.9191180.

25. C. Dorea and R. L. de Queiroz, "Adaptive Block Partitioning of Point Clouds for Video-Based Color Compression," 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 2020, pp. 2746-2749, doi: 10.1109/ICIP40778.2020.9191065.