ISSN: 1533 - 9211
https://seyboldreport.net/

# HIGH-FIDELITY GENERATIVE IMAGE COMPRESSION USING GAN'S AND XGBOOST APPROACH

**Gadde Suresh [1], K.S.Thivya [2] & M.Anand [3]**
Research scholar [1], Professor [2&3]
Department of Electronics and Communication Engineering,
Dr. MGR. Educational and Research Institute , Chennai 95.

**Abstract**
Our research delved into the integration of Generative Adversarial Networks, XG Boost, and compression techniques to develop a sophisticated generative lossy compression system. The investigation covered various factors such as normalization layers, generator and discriminator architectures, training strategies, and perceptual losses. Our system produced visually appealing reconstructions similar to the original input, functioning effectively across a wide range of bitrates and even for high-resolution images. We evaluated the system's performance using several perceptual metrics and a user study, demonstrating our approach to be superior to existing methods, even at bitrates exceeding 2 x bitrate. In summary, our research successfully bridged the gap between rate-distortion-perception theory and practical implementation.

## 1Introduction

The usage of images is growing at a staggering pace, leading to an exponential rise in the storage space and bandwidth required to store and transmit them. Consequently, there has been considerable attention given to image compression methods that can eliminate redundant information and reduce data size, thereby enabling more efficient storage and transmission. The two primary digital image compression techniques are lossless and lossy. In lossless compression, the retrieved information from the compressed image is identical to the original information before compression. On the other hand, lossy compression produces similar but not identical information. However, lossy compression techniques can considerably decrease the size of the compressed image and fine-tune the balance between the produced data size and the retrieved image quality when compared to the original. As a result, several recent methods have emerged that compress the same image at various compression levels based on the significance of the available information in the image.

Recent advancements in Artificial Neural Networks (ANNs), machine learning, and deep learning have led to significant improvements in image compression performance. These techniques offer greater flexibility in terms of the type of objects in the images being compressed. Furthermore, adding new image types to the compression process only requires training the discriminator to accurately identify whether the generated image matches the original, as these networks are not limited by hand-crafted features and can learn new ones through additional training. In this study, we propose a deep learning approach that leverages GAN and XGBoost to ensure that the compressed image retains the visual appeal of the original. We also employ a training method that prevents the discriminator from being fooled by the generator, thus preserving the clarity of the compressed image.

New research has shown that attention mechanisms can be a valuable addition to Generative Adversarial Networks (GANs) for image compression tasks [7], [8]. By breaking down the generative network into separate attention and transformation networks, the attention network can identify key areas of interest, while the transformation network converts the image from one domain to another. In [7] and [8], the CycleGAN framework is enhanced with additional attention networks to preserve the background of the input image while transforming the foreground. This approach has demonstrated improved performance for image compression with GANs.

As our reliance on image grows, the demand for storage space and bandwidth to accommodate them is becoming increasingly unfeasible. This has prompted the need for compression techniques that can reduce the number of bits needed to represent an image without sacrificing its quality. Image compression plays a crucial role in numerous applications by significantly decreasing storage space and communication bandwidth requirements, making it easier to deploy imaging technologies at scale.

This paper introduces a novel approach for compressing high-resolution images while preserving their visual quality. The proposed methodology is compared to existing methods, and the results show that the proposed approach outperforms them visually, even when the previous methods used higher bitrates. The study discusses various quantitative metrics to evaluate the approach's performance such as KID,FID,PSNR,LPIPS,MSE and SSIM which demonstrate that the results align with the rate-distortion-perception theory. Although no single metric can precisely predict the user study's exact ranking, metrics such as FID and KID can be beneficial in guiding the exploration process and provide useful insights for informed decision-making. A comprehensive perceptual evaluation requires a diverse range of metrics that cover different aspects, including no-reference metrics, pair-wise similarities, distributional similarities, and deep feature-based metrics derived from various network architectures. By employing this ensemble of metrics, a more robust and comprehensive evaluation of perceptual qualities can be achieved. The analysis thoroughly examines the suggested architecture and its components, such as normalization layers, generator and discriminator architectures, training methods, and the loss function, based on both perceptual metrics and stability to obtain a comprehensive understanding of their effectiveness.

## 2 Related Work

Deep convolutional autoencoders have demonstrated significantly better compression rates than existing techniques like JPEG2000 with similar complexity [13]. This has led to the use of neural networks in various approaches to tailor their performance for specific applications. Ayoobkhan et al. [7] proposed a hybrid compression method that uses near-lossless compression for the ROI and lossy compression for the remaining parts of the image. Their method utilizes Graph-Based Segmentation (GBS) [17] to extract the ROI, and a Feed-Forward Neural Network (FF-NN) is trained on the GBS segments to compress image data. The FF-NN provides near-lossless predictions of pixel values based on their neighbouring pixels' values. The method optimizes edge weight values using two optimization algorithms: Particle Swarm (PS) and Gravitational Search (GS) optimizers. Although this method performs well compared to previous methods, a Convolutional Neural Network (CNN) can achieve better results by utilizing three-dimensional filters that consider pixel position in addition to their values, providing superior performance over FF-NN when interacting with images.

In the beginning, RNNs were utilized in the initial works [45, 47], whereas subsequent works were based on auto-encoders [5, 44, 1]. To achieve a reduced bitrate, various approaches have been employed to enhance the modelling of the probability density of auto-encoder latents, which, in turn, leads to more efficient arithmetic coding. These methods include hierarchical priors, auto-regression with different context shapes, or a combination of both [6, 31, 28, 39, 32, 26, 33]. State-of-the-art models, such as the one proposed by Minnen et al. [32], now surpass BPG in terms of PSNR.

A novel spatial attention GAN model introduced by Hajar Emami et al. [59], have facilitated significant advancements in the field of image-to-image translation.The discriminator in SPA GAN computes attention and leverages it to guide the generator to focus on the most important regions that distinguish between the source and target domains. The attention is represented by spatial maps, which highlight the areas that the discriminator considers significant in determining whether an input image is real or fake. These spatial attention maps are then fed back to the generator, resulting in higher emphasis on the discriminative regions when computing the generator loss, thereby producing more realistic output images.

CNNs have been successfully applied to compression tasks using autoencoders, which typically comprise encoding, bottleneck, and decoding layers. The encoding layers down sample the input until the bottleneck layer, the smallest in the network, is reached. The decoding layers then restore the input size until the output layer. During training, the network minimizes the difference between the output and input images, aiming to reconstruct the original image from the bottleneck layer. After training, the encoding network generates the compressed image, which can be used with the decoding network to retrieve the original. CNNs also excel at segmenting images by predicting pixel segmentation. Among the most effective networks for medical image segmentation is the U-net, which has shown excellent performance [12].

## 3 Method

### 3.1 Background

**Conditional GANs:**

Conditional GANs are a type of generative adversarial network where the generator is conditioned on additional information, such as labels or input images. The additional information is usually fed into the generator and discriminator networks as extra inputs. The aim of the network is to learn a mapping between the additional information and the generated images, while the discriminator tries to distinguish between real and fake images. The generator tries to generate images that are realistic and similar to the target distribution while satisfying the conditions specified in the additional information. These networks have found various applications, such as image-to-image translation, super-resolution, and text-to-image synthesis.

Conditional Generative Adversarial Networks (GANs) are a machine learning method that enables the learning of a generative model of a conditional distribution p(X|S), where S represents additional information or context associated with a given data point X, such as class labels or semantic maps. The joint distribution p (X,S) is often unknown, and Conditional GANs help in estimating it. The method has been widely used in various applications to generate images and other data that are conditioned on specific contexts. In Conditional GANs, two networks, a generator G and a discriminator D, are trained to learn a generative model of a conditional distribution p (X|S). The generator G, dependent on the information s, transforms samples y from a fixed known distribution pY into p(X|S). The discriminator D receives (x, s) input and evaluates the probability of it being a sample from p (X|S) rather than from G's output. The goal is to train the generator G to generate samples that can deceive the discriminator D into classifying them as real data coming from the distribution p(X|S). To achieve this, a "non-saturating" loss can be optimized while keeping s constant during the process [17,58,59].

$$V_{DG} = E_{X \sim} P_{data(x)} [\log D (X)] + E_{Z -} P_{data\,(z)} [\log (1 - D (G (Z))] \qquad (1)$$

G- generator, X- sample from real data, Z- sample from generator, D- Discriminator, $P_{data(x)}$- distribution of real data, $P_{data\,(z)}$- distribution of generator data, D(x) – Discriminator network and G(x)- generator network

**XGBoost:**

XGBoost is a popular machine learning algorithm that has been widely used for regression and classification tasks. It is an implementation of gradient boosted decision trees that is designed to be efficient, scalable, and flexible. The algorithm has several advantages over other machine learning methods, such as faster execution times, better accuracy, and the ability to handle large datasets.

In XGBoost, the objective is to minimize a loss function that measures the difference between the predicted and actual values. The loss function is defined as the sum of two terms: the first term is the training loss, which measures the difference between the predicted and actual values for each training example, and the second term is the regularization term, which helps to prevent overfitting.

The algorithm builds a model by iteratively adding decision trees to the ensemble, with each subsequent tree trying to correct the errors of the previous tree. The trees are constructed in a greedy manner, using a split finding algorithm that maximizes the reduction in the loss function at each step.

The XGBoost algorithm also includes several techniques to improve its performance and reduce overfitting, such as pruning, regularization, and early stopping. Pruning involves removing branches from the tree that do not contribute to improving the overall performance, while regularization adds penalties to the loss function to discourage overfitting. Early stopping is used to stop the training process if the performance of the model does not improve after a certain number of iterations.

Mathematically, the objective function for XGBoost can be written as:

$$Obj = L + \Omega \quad (2)$$

where L is the training loss function and Ω is the regularization term. The training loss function can be written as:

$$L = \sum(yi - \hat{y}i)^2 \quad (3)$$

where yi is the true label of the i-th example and ŷi is the predicted label. The regularization term can be written as:

$$\Omega = \gamma T + \tfrac{1}{2}\lambda \sum wi^2 \quad (4)$$

where T is the number of leaves in the tree, γ is the complexity parameter that controls the size of the tree, λ is the regularization parameter that controls the amount of regularization, and wi is the weight assigned to the i-th feature.

**Neural Image Compression**

Neural image compression is a technique that uses deep learning models to reduce the size of digital images while maintaining their visual quality. The approach involves training an encoder-decoder network, also known as an autoencoder, to compress and decompress the images. The encoder converts the high-dimensional image data into a lower-dimensional representation, while the decoder generates a compressed version of the image from this representation. The goal is to minimize the distortion between the original and the compressed image while constraining the compression rate. This can be achieved by adding a rate-distortion trade-off term to the loss function that balances the level of compression with the amount of distortion allowed in the compressed image. Neural image compression has many practical applications, including reducing the storage and transmission requirements of large image datasets without sacrificing image quality.

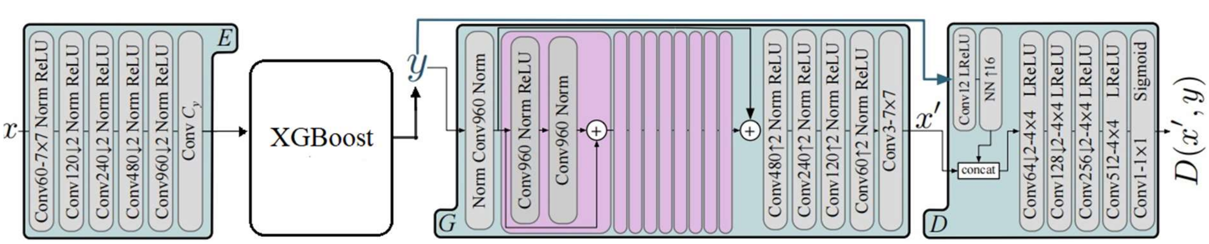
Shannon's rate-distortion theory is the foundation for the concept of learned lossy compression, which involves balancing the amount of information (rate) with the acceptable level of distortion during compression. The common approach to this problem is through an auto-encoder architecture comprising two components: an encoder (E) and a decoder (G). In learned lossy compression, the auto-encoder quantizes an image x into a latent representation y = E(x) and uses the decoder G to reconstruct the image x' from y. The distortion incurred during lossy compression is measured using metrics like mean squared error (MSE). To store the quantized latent y, a probability model P is introduced, and an entropy coding algorithm such as arithmetic coding is used to store y losslessly. The bitrate required to store y can be calculated as r(y) = -log(P(y)), with the entropy coder incurring some overhead bits. By parameterizing E, G, and P as convolutional neural networks (CNNs), they can be simultaneously trained to optimize the trade-off between rate and distortion, with a parameter ζ controlling the balance. Shannon's rate-distortion theory [14, 58] serves as the basis for learned lossy compression.

$$V_{EG} = E_{X\sim Px}\,[\zeta\, r(y) + d\,(x, x')] \tag{5}$$

### 3.2 Formulation and Optimization

To achieve neural image compression, we combine a conditional GAN with a learned lossy compression approach. This involves using an auto-encoder with an encoder E and a decoder G to quantize an image x into a latent representation y, and then storing y in a lossless manner, the XGBoost classifier helps to predict the neighbour in the creation of the sample copy by the generator. We train E and G, along with a discriminator D and a generator K, to minimize a trade-off between the compression rate and the distortion of the reconstructed image, as measured by a combined loss term d that includes both mean squared error and a perceptual distortion metric called LPIPS. By tuning hyperparameters, including ζ, ح, kM, and kP, we can optimize the trade-off and achieve superior compression results.

$$L_{EGK} = E_{x\sim px}\,[\,\zeta\, r(y) + d\,(x,x') - \text{ح}\log D\,(x',y)] \tag{6}$$




In Figure 1, we present our architecture, where ConvC denotes a convolution operation with C channels using 3x3 filters, unless otherwise specified. Strided down or up convolutions are denoted by ↓2 and ↑2, respectively. For normalization, we use ChannelNorm as described in the text. The activation function we use is the leaky ReLU with α=0.2, as defined in previous studies [56,58]. We employ nearest neighbor interpolation with a factor of 16, denoted as NN↑16, for upsampling. Lastly, we use Q quantization as described in [58].

**Constrained Rate**

When training a neural compression model using the loss function mentioned in Equation 5, regulating the ultimate bitrate is typically done by manipulating ζ. However, our context involves multiple conflicting terms (MSE, dP, and -log(D(x'))) with the rate term, making it challenging to compare models with varying hyperparameters kM, kP, and ج when ζ is kept constant. To address this issue, we introduce a "rate target" hyperparameter (rt) and modify Equation 3 by replacing ζ with an adaptive term, ζ'. The value of ζ' depends on two additional hyperparameters, ζ(a) and ζ(b), and is set to ζ(a) if r(y) is greater than rt and ζ(b) otherwise. By setting ζ(a) much greater than ζ(b), we can train models with an average bitrate close to rt. This approach helps to ensure that models are optimized for a specific bitrate target while still considering multiple conflicting terms.

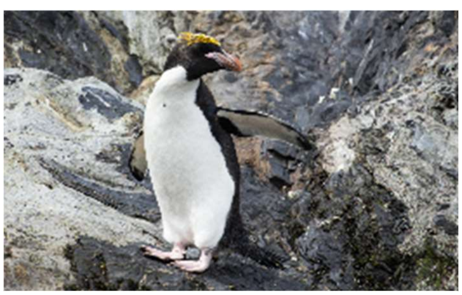
## 3.3 Architecture

Figure 1 depicts our architecture, comprising of the encoder E, generator G, discriminator D and XGBoost classifier block. We adopt the straight-through estimator, as in [44,58], for rounding y before inputting it into G. While our E, G, and D are based on [51,58,3], we introduce several distinctive modifications in the discriminator and normalization layers, which will be described in detail in the upcoming sections. While both [51,58, 3] utilize a multi-scale patch-discriminator D, we implement a single-scale patch-discriminator D and replace InstanceNorm [49,58] with SpectralNorm [36,58]. In contrast to [3], we condition D on y by concatenating an upscaled version of it with the image, as illustrated in Figure 1 [58]. This approach is inspired by the use of a conditional GAN formulation associated with XGBoost, where D can access the conditioning information (in our case, y as described in Section 3.2).

# 4 Experiments

In Figure 1, we present our architecture, which consists of the encoder E, generator G, discriminator D, and XGBoost classifier block. To round y, we use the straight-through estimator, similar to [44,58]. While our E, G, and D are based on [51,58,3], we introduce some unique modifications in the discriminator and normalization layers, which we will describe in detail in later sections. Unlike [51,58,3], we employ a single-scale patch-discriminator D and substitute InstanceNorm [49,58] with SpectralNorm [36,58]. In addition, we condition D on y by concatenating an upscaled version of it with the image, as shown in Figure 1 [58]. This strategy is motivated by the use of a conditional GAN formulation with XGBoost, where D can access the conditioning information (in our case, y as described in Section 3.2).

| Original Image | Original size | Hi-fi-Gan’s+ XGBoost High | Hi-fi-Gan’s+ XGBoost Medium | Hi-fi-Gan’s+ XGBoost Low |
|---|---|---|---|---|
| | 4.3 MB | 1.5 MB | 1.6MB | 1.6 MB |
| | | Hi-fi-Gan’s+KNN | Hi-fi-Gan’s+KNN | Hi-fi-Gan’s+KNN |

| | | | | |
|---|---|---|---|---|
| <br>Image 1<br>Resolution: 2048*1366 | | High | Medium | Low |
| | | 1.9 MB | 1.8MB | 1.9 MB |
| | | Hi-fi-high | Hi-fi-medium | Hi-fi-low |
| | | 2.8 MB | 2.0 MB | 2.5 MB |
| <br>Image 2<br>Resolution:2016*1562 | 3.5 MB | Hi-fi-Gan's+ XGBoost High | Hi-fi-Gan's+ XGBoost Medium | Hi-fi-Gan's+ XGBoost Low |
| | | 1.3 MB | 1.5 MB | 1.4 MB |
| | | Hi-fi-Gan's+KNN High | Hi-fi-Gan's+KNN Medium | Hi-fi-Gan's+KNN Low |
| | | 1.5 MB | 1.6 MB | 1.6 MB |
| | | Hi-fi-high | Hi-fi-medium | Hi-fi-low |
| | | 1.9 MB | 2.0 MB | 2.1 MB |
| <br>Image 3<br>Resolution:2040*1356 | 4.5 MB | Hi-fi-Gan's+ XGBoost High | Hi-fi-Gan's+ XGBoost Medium | Hi-fi-Gan's+ XGBoost Low |
| | | 2.3 MB | 2.5 MB | 2.3 MB |
| | | Hi-fi-Gan's+KNN High | Hi-fi-Gan's+KNN Medium | Hi-fi-Gan's+KNN Low |
| | | 2.7 MB | 2.7 MB | 2.8 MB |
| | | Hi-fi-high | Hi-fi-medium | Hi-fi-low |
| | | 3.4 MB | 3.7 MB | 3.7 MB |
| | 3.9 MB | Hi-fi-Gan's+ | Hi-fi-Gan's+ | Hi-fi-Gan's+ |

| | | | | |
|---|---|---|---|---|
| <br>Image 4<br>Resolution:2040*1356 | | XGBoost High | XGBoost Medium | XGBoost Low |
| | | 1.1 MB | 1.3 MB | 1.3 MB |
| | | Hi-fi-Gan’s+KNN High | Hi-fi-Gan’s+KNN Medium | Hi-fi-Gan’s+KNN Low |
| | | 1.4 MB | 1.4 MB | 1.4 MB |
| | | Hi-fi-high | Hi-fi-medium | Hi-fi-low |
| | | 2.3 MB | 2.6 MB | 2.7 MB |
| <br>Image 5<br>Resolution:2040 *1536 | 3.4 MB | Hi-fi-Gan’s+ XGBoost High | Hi-fi-Gan’s+ XGBoost Medium | Hi-fi-Gan’s+ XGBoost Low |
| | | 600 KB | 617.5 KB | 620.2i KB |
| | | Hi-fi-Gan’s+KNN High | Hi-fi-Gan’s+KNN Medium | Hi-fi-Gan’s+KNN Low |
| | | 623.9 KB | 617.5 KB | 638.0 KB |
| | | Hi-fi-high | Hi-fi-medium | Hi-fi-low |
| | | 1.0 MB | 1.2 MB | 1.3 MB |
| <br>Image 6<br>Resolution:2040*1200 | 4.1 MB | Hi-fi-Gan’s+ XGBoost High | Hi-fi-Gan’s+ XGBoost Medium | Hi-fi-Gan’s+ XGBoost Low |
| | | 1.0 MB | 1.3 MB | 1.1 MB |
| | | Hi-fi-Gan’s+KNN High | Hi-fi-Gan’s+KNN Medium | Hi-fi-Gan’s+KNN Low |
| | | 1.7 MB | 1.8 MB | 1.8 MB |
| | | Hi-fi-high | Hi-fi-medium | Hi-fi-low |
| | | 2.6 MB | 2.9 MB | 2.9MB |
| | 3.7 MB | Hi-fi-Gan’s+ | Hi-fi-Gan’s+ | Hi-fi-Gan’s+ |

| | | | | |
|---|---|---|---|---|
|  Image 7 Resolution: 2040*1152 | | XGBoost High | XGBoost Medium | XGBoost Low |
| | | 1.0 MB | 1.2 MB | 1.1 MB |
| | | Hi-fi-Gan's+KNN High | Hi-fi-Gan's+KNN Medium | Hi-fi-Gan's+KNN Low |
| | | 1.4 MB | 1.4 MB | 1.4 MB |
| | | Hi-fi-high | Hi-fi-medium | Hi-fi-low |
| | | 2.5 MB | 2.4 MB | 2.4 MB |

Table 1. Compressed size is showcased for Hi-fi-Gan's+ XGBoost high, medium and low approach vs Hi-fi-Gan's+KNN high, medium and low approach Vs Hi-fi-high, medium and low approach

| COMPRESSED IMAGE VISUAL CLARITY | | |
|---|---|---|
| Hi-fi-Gan's+ XGBoost High | Hi-fi-Gan's+ XGBoost Medium | Hi-fi-Gan's+ XGBoost Low |
|  Image 1 |  |  |
|  Image 2 |  |  |

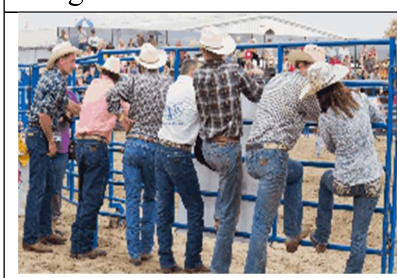
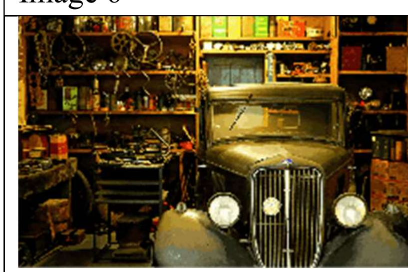
| | | |
|---|---|---|
|  Image 3 |  |  |
|  Image 4 |  |  |
|  Image 5 |  |  |
|  Image 6 |  |  |
|  Image 7 |  |  |

Table 2: Compressed Image visual quality is showcased for Hi-fi-Gan's+XGBoost high, medium and low approach.

| Hi-fi-Gan's+ XGBoost Low | | | | | | |
|---|---|---|---|---|---|---|
| Sample Image | FID | KID | LPIPS | PSNR | MSE | SSIM |
| Image 1 | 78.454 | 5.53026 | 4.60286 | 32.413 | 50.30 | 0.87 |
| Image 2 | 140.567 | 9.6711 | 8.0748 | 32.144 | 60.30 | 0.86 |
| Image 3 | 101.115 | 7.0410 | 5.8695 | 31.228 | 70.12 | 0.86 |
| Image 4 | 56.703 | 4.0802 | 3.3870 | 31.766 | 30.74 | 0.80 |
| Image 5 | 61.244 | 4.3826 | 3.6408 | 30.438 | 38.13 | 0.88 |
| Image 6 | 150.393 | 10.3262 | 8.6240 | 30.391 | 50.12 | 0.83 |
| Image 7 | 106.548 | 7.4032 | 6.1732 | 31.895 | 50.10 | 0.88 |

(A)

| Hi-fi-Gan's+ XGBoost Medium | | | | | | |
|---|---|---|---|---|---|---|
| Sample Image | FID | KID | LPIPS | PSNR | MSE | SSIM |
| Image 1 | 47.556 | 3.4704 | 2.8754 | 32.220 | 38.19 | 0.90 |
| Image 2 | 92.930 | 6.4953 | 5.4120 | 32.612 | 50.52 | 0.90 |
| Image 3 | 58.364 | 4.1909 | 3.4798 | 32.236 | 45.73 | 0.87 |
| Image 4 | 42.757 | 3.1504 | 2.6075 | 32.096 | 29.19 | 0.84 |
| Image 5 | 58.580 | 4.2053 | 3.4919 | 30.478 | 30.21 | 0.93 |
| Image 6 | 102.988 | 7.1658 | 5.9742 | 30.762 | 40.37 | 0.88 |
| Image 7 | 72.770 | 5.1513 | 4.2851 | 30.321 | 44.13 | 0.87 |

(B)

| Hi-fi-Gan's+ XGBoost High | | | | | | |
|---|---|---|---|---|---|---|
| Sample Image | FID | KID | LPIPS | PSNR | MSE | SSIM |
| Image 1 | 33.501 | 2.5334 | 2.0901 | 30.71 | 25.16 | 0.91 |
| Image 2 | 57.837 | 4.1558 | 3.4504 | 30.05 | 40.12 | 0.91 |
| Image 3 | 38.063 | 2.8375 | 2.3451 | 30.81 | 30.19 | 0.94 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Image 4 | 38.481 | 2.8654 | 2.3684 | 31.32 | 29.12 | 0.95 |
| Image 5 | 57.833 | 4.1555 | 3.4502 | 31.33 | 29.10 | 0.95 |
| Image 6 | 77.084 | 5.4389 | 4.5262 | 31.45 | 27.32 | 0.96 |
| Image 7 | 59.002 | 4.2334 | 3.5155 | 30.05 | 40.12 | 0.87 |

(C)

Table 3. A,B,C, Test result for proposed system for different images

It is evident from table 3, that Hi-fi-Gan's+XG Boost high approach produces higher structural similarity index measure and lesser Mean square error when compared to Hi-fi-Gan's+XGBoost low and medium approach. The MSE decreases as the resolution increases and PSNR increases as the resolution improves. FID, or Fréchet Inception Distance, is a metric that is frequently used to assess the quality of computer-generated images in comparison to real images. This metric is based on the Fréchet distance, which measures the distance between the feature representations of the generated images and those of the real images, both of which are obtained from the Inception v3 network. When the FID score is lower, it indicates that the generated images are more similar to real images. FID is commonly used to evaluate the performance of generative models such as GANs.

KID, or Kernel Inception Distance, is another metric used to evaluate the quality of computer-generated images compared to real images. KID is based on the maximum mean discrepancy (MMD) between the feature representations of the generated and real images, both of which are obtained from the Inception v3 network. KID is considered to be more robust to noise than FID. Like FID, a lower KID score indicates that the generated images are more similar to real images.

LPIPS, or Learned Perceptual Image Patch Similarity, is a metric that measures the perceptual similarity between two images. This metric is based on the idea that the similarity between the feature representations of two images can be used to quantify their perceptual similarity. The feature representations are obtained from a pre-trained deep neural network, and LPIPS is trained to be perceptually linear. It is capable of capturing both global and local perceptual differences between two images. LPIPS has been shown to be highly correlated with human perceptual judgments and is widely used in tasks such as image quality assessment and image-to-image translation. MSE, SSIM, and PSNR are all metrics used to evaluate the quality and similarity of two images. MSE, which stands for Mean Squared Error, is calculated by finding the average of the squared differences between each pixel value of the two images. A lower MSE score indicates that the images are more similar to each other. SSIM, or Structural Similarity Index, takes into account differences in luminance, contrast, and structure between the two images. A higher SSIM score indicates that the images are more similar in terms of their structure. PSNR, or Peak Signal-to-Noise Ratio, measures the quality of an image by comparing it to a reference image. It calculates the ratio between the maximum possible pixel value and the root-mean-square error (RMSE) between the two images. A higher PSNR value indicates that the image quality is better.

**Conclusion:**

In this research, we propose a novel compression technique called High-Fidelity Generative Image Compression Using GAN's and XGBoost. The technique has three different variations: Hi-fi-GAN's+XGBoost High, Hi-fi-GAN's+ XGBoost Medium, and Hi-fi-GAN's+XGBoost Low. The paper discusses the architecture of our proposed system, which includes an encoder E, generator G, discriminator D, and XGBoost classifier block. Our approach has better performance and compression size, and the compressed image is clearer compared to two approaches (i) that solely uses GAN, called "High-Fidelity Generative Image Compression," in three variations: Hi-fi-low, Hi-fi-medium, and Hi-fi-high & (ii) High-Fidelity Generative Image Compression Using GAN's and KNN- high, medium and low approach. In our technique, GAN works on two parameters: x and s. x represents data point or picture location, and s represents extra information from absolute location about features. We fixed x as 7*7 (49 pixels) to avoid multiple sizes of samples generated by the generator G. This approach reduces the permutation and increases accuracy by training only one single location of the sample, 7*7 (49 pixel), instead of the whole image. We fixed Y, which represents the total number of samples, as 120, 240, 480, and 960, to avoid non-saturating loss. We employed rectified linear unit and Leaky ReLU in the discriminator part to improve accuracy. Our proposed architecture outperforms the High-Fidelity Generative Image Compression technique with Gan’s network and High-Fidelity Generative Image Compression technique with Gan’s and KNN approach.

**References**

[1] Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc V Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. In *Advances in Neural Information Processing Systems*, pages 1141–1151, 2017.
[2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.
[3] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
[5] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.
[6] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations (ICLR)*, 2018.

[7] Fabrice Bellard. BPG Image format. https://bellard.org/bpg/.
[8] Mikołaj Binkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018.
[9] Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. *arXiv preprint arXiv:1901.07821*, 2019.
[10] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fifidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.
[11] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
[12] Rémi Coulom. Whole-history rating: A bayesian rating system for players of time-varying strength. In *International Conference on Computers and Games*, pages 113–124. Springer, 2008.
[13] Stéphane Coulombe and Steven Pigeon. Low-complexity transcoding of jpeg images with near-optimal quality using a predictive quality factor and scaling parameters. *IEEE Transactions on Image Processing*, 19(3):712–721, 2009.
[14] Thomas M Cover and Joy A Thomas. Elements of information theory. John Wiley & Sons, 2012.
[15] Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. Deep generative adversarial compression artifact removal. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4826–4835, 2017.
[16] Mark E Glickman. A comprehensive guide to chess ratings. American Chess Journal, 3(1):59–102, 1995.
[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.
[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Advances in Neural Information Processing Systems, pages 6626–6637, 2017.
[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning, pages 448–456, 2015.
[20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In International Conference on Learning Representations (ICLR), 2017.
[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4401–4410, 2019.
[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2014.

[23] Kodak PhotoCD dataset. http://r0k.us/graphics/kodak/.
[24] Karol Kurach, Mario Lucic, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. A large-scale study on regularization and normalization in gans. In International Conference on Machine Learning, pages 3581–3590, 2019.
[25] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image superresolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4681–4690, 2017.
[26] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. arXiv preprint arXiv:1809.10452, 2018.
[27] Boyi Li, Felix Wu, Kilian Q Weinberger, and Serge Belongie. Positional normalization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 1622–1634. Curran Associates, Inc., 2019.
[28] Mu Li, Wangmeng Zuo, Shuhang Gu, Debin Zhao, and David Zhang. Learning convolutional networks for content-weighted image compression. arXiv preprint arXiv:1703.10553, 2017.
[29] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In Advances in Neural Information Processing Systems, pages 700–709, 2018.
[30] Detlev Marpe, Heiko Schwarz, and Thomas Wiegand. Context-based adaptive binary arithmetic coding in the h. 264/avc video compression standard. IEEE Transactions on circuits and systems for video technology, 13(7):620–636, 2003.
[31] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[32] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. In Advances in Neural Information Processing Systems, pages 10771–10780, 2018.
[33] David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. arXiv preprint arXiv:2007.08739, 2020.
[34] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.
[35] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. IEEE Signal Processing Letters, 20(3):209–212, 2012.
[36] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In International Conference on Learning Representations, 2018.
[37] Sahand Negahban, Sewoong Oh, and Devavrat Shah. Iterative ranking from pair-wise comparisons. In Advances in neural information processing systems, pages 2474–2482, 2012.
[38] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image

synthesis with spatiallyadaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.
[39] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2922–2930, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.
[40] Shibani Santurkar, David Budden, and Nir Shavit. Generative compression. *arXiv preprint arXiv:1703.01467*, 2017.
[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
[42] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high effificiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649– 1668, 2012.
[43] David S. Taubman and Michael W. Marcellin. *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.
[44] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszar. Lossy image compression with compressive autoencoders. In *International Conference on Learning Representations (ICLR)*, 2017.
[45] George Toderici, Sean M O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. *arXiv preprint arXiv:1511.06085*, 2015.
[46] George Toderici, Lucas Theis, Nick Johnston, Eirikur Agustsson, Fabian Mentzer, Johannes Ballé, Wenzhe Shi, and Radu Timofte. *CLIC 2020: Challenge on Learned Image Compression*, 2020. http:// compression.cc.
[47] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. *arXiv preprint arXiv:1608.05148*, 2016.
[48] Michael Tschannen, Eirikur Agustsson, and Mario Lucic. Deep generative models for distributionpreserving lossy compression. In *Advances in Neural Information Processing Systems*, pages 5929–5940, 2018.
[49] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.
[50] Gregory K Wallace. The JPEG still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.
[51] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Highresolution image synthesis and semantic manipulation with conditional gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
[52] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems Computers, 2003*, volume 2, pages 1398–1402 Vol.2, Nov 2003.

[53] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.
[54] WebP Image format. https://developers.google.com/speed/webp/.
[55] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
[56] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.
[57] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.
[58] Fabian Mentzer, George Toderici,Michael Tschannen, Eirikur Agustsson. High-Fidelity Generative Image Compression. In 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada. arXiv:2006.09965v3 [eess.IV] 23 Oct 2020.
[59] Hajar Emami, Majid Moradi Aliabadi, Ming Dong, and Ratna Babu Chinnam, SPA-GAN: Spatial Attention GAN for Image-to-Image Translation. arXiv:1908.06616v3 [cs.CV] 30 Dec 2020.
[60] Gadde Suresh , K.S.Thivya & M.Anand , High-Fidelity Generative Image Compression Using Gan's And KNN Approach, in Journal of Data Acquisition and Processing, ISSN: 1004-9037 https://sjcjycl.cn/ DOI: 10.5281/zenodo.776810