JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Image Compression using Generative Adversarial Network

[1]Chitra Wasnik, [2]Prathamesh Tamanekar, [3]Nishant Joshi, [4]Shubham Deokar
[1]Professor, [2]Student, [3]Student, [4]Student
[1]Computer Science and Engineering (Artificial Intelligence and Machine Learning), [2-4]Computer Engineering
[1-4]Lokmanya Tilak College of Engineering, Navi Mumbai, India

Abstract: Over the past decade, due to the exponential rise in the number of internet and computer users, a colossal amount of data is transferred and used on a daily basis. To enable such a magnitude of data transfer quickly and seamlessly, data compression has become a very crucial part of the mechanism. Thus, we aim to aid and improve the pre-existing image compression algorithms using Artificial Intelligence and Deep learning. The focus of this research and development efforts is on increasing the algorithm speed and compression ratio while achieving a near-lossless compression. We strive to program a neural network for compressing an image while allowing a threshold data loss and later enhancing the image to make up for the lost data, this approach will allow us to reduce the time required for compressing a file as lossy compression is carried out. At the same time, the image enhancement engine manages the data loss and keeps it to a minimum. The applications of such algorithms are infinite, ranging from image transfer, video streaming, and video calls to enhance the cloud experience.

***IndexTerms* – Image Compression, Image Processing, Artificial Intelligence, Neural Networks, Generative Adversarial Network.**

I. INTRODUCTION

An Image Compression tool that will allow the user to transfer or convert images that will be enhanced and compressed with respect to the images uploaded initially. This will be possible using Artificial Intelligence (AI) for Image Generation, Manipulation, and Enhancement. For Image Generation “Generative Adversarial Network” (GAN) is used. GANs is an approach to generative modelling using deep learning methods, such as convolutional neural networks. Generative modelling is an unsupervised learning task in machine learning that involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset. GANs are a clever way of training a generative model by framing the problem as a supervised learning problem with two sub-models: the generator model that we train to generate new examples, and the discriminator model that tries to classify examples as either real (from the domain) or fake (generated). The two models are trained together in a zero-sum game, adversarial, until the discriminator model is fooled about half the time, meaning the generator model is generating plausible examples.

Due to improved computational power as well as improved and diversified machine learning and deep learning techniques, they have been explored and applied in many high-level image processing applications such as object detection, pattern recognition, optical character recognition, classification, speech recognition, face recognition, computer vision, super-resolution, image inpainting, image generation, image enhancement, machine translation, etc. However, it is limitedly used in image compression kind of low-level image processing in recent years only.

Data compression has become a common requirement for most application software as well as an important and active research area in computer science. Without compression techniques, none of the ever-growing Internet, digital TV, mobile communication, or increasing video communication techniques would have been practical developments.

The main focus is to increase the algorithm speed and compression ratio while achieving a near-lossless compression. The various applications of this algorithm are Image transfer, Video Streaming, Video calls, Cloud Computing, Astrophysical images, medical imaging, etc.

II. Related Works

The existing systems for image compression that uses Artificial Intelligence and Neural Networks are primarily based upon an encoder-decoder model where convolutional neural networks are used for compressing an image before the image is passed through an encoder and another neural network is established to reconstruct the image after decoding the image.

All methods except the GAN model (BPG, Rippel et al., and Minnen et al.) employ adaptive arithmetic coding using context models for improved compression performance. Such models could also be implemented for our system, and have led to additional savings of 10% in [6]. Since Rippel et al. and Minnen et al. have only released a selection of their decoded images, and at significantly higher bitrates, a comparison with a user study is not meaningful. Instead, we try to qualitatively put our results into context with theirs. We can observe that even though Rippel et al. [9] use 29–179% more bits, GAN models produce images of comparable or better quality. First, we see that BPG is still visually competitive with the current state-of-the-art, which is consistent with moderate 8.41% bitrate
savings being reported by [8] in terms of PSNR. Second, even though we use much fewer bits compared to the example images available from [8], for some of them our method can still produce images of comparable visual quality.

Table 1 Comparison of related works

Sr. No	Authors	Title of the paper & year of publ.(Old to recent)	Major contributions
1	Shinobu Kudo Shota Orihashi Ryuichi Tanida Atsushi Shimizu	GAN-based Image Compression Using Mutual Information Maximizing Regularization (2019)	Importance of regularization.
2	E. Agustsson M. Tschannen F. Mentzer, R. Timofte L. V. Gool	Generative Adversarial Networks for Extreme Learned Image Compression (2018)	Implementation of GAN for image compression
3	O. Rippel L. Bourdev	Real-Time Adaptive Image Compression (May 2017)	Adaptive Image compression.
4	I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville Y. Bengio	Generative Adversarial Nets. (Dec. 2014)	Use of GAN (Generative Adversarial networks)

It is a dual network. The first network, will take the image and generate a compact representation (ComCNN). The output of this network will then be processed by a standard codec (e.g. JPEG). After going through the codec, the image will be passed to a 2nd network, which will 'fix' the image from the codec, trying to get back the original image. The authors called it Reconstructive CNN (RecCNN). Both networks are iteratively trained, similar to a GAN.

The figure below demonstrates the existing model for image compression.

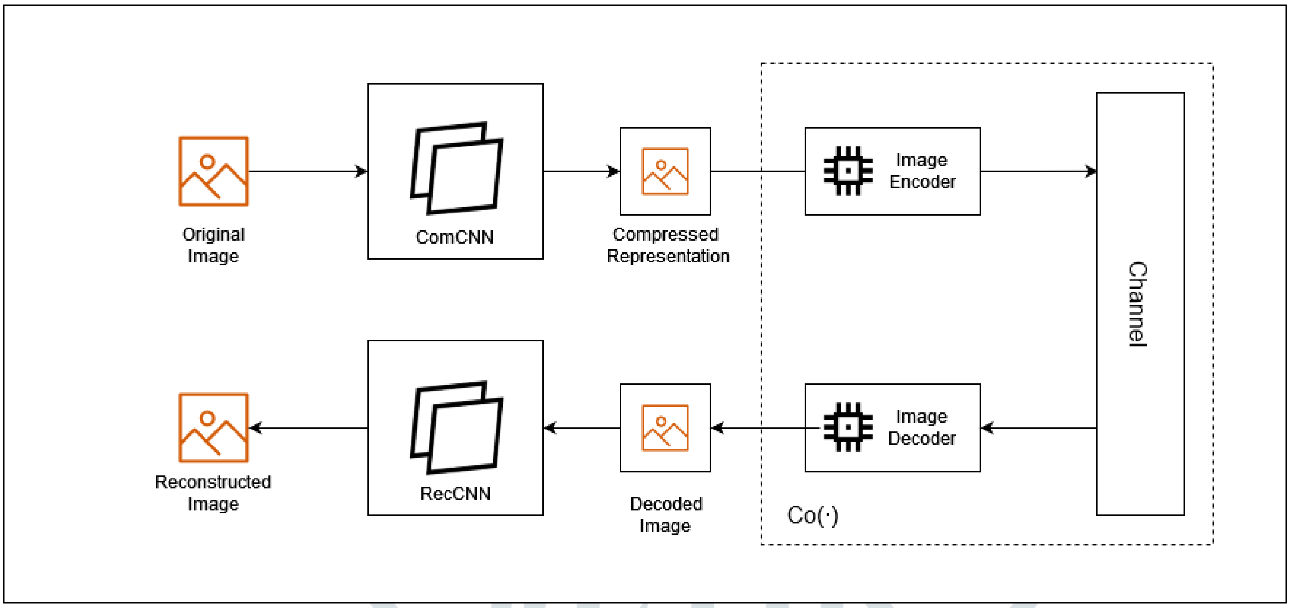

Figure 1 Architecture of existing system.

III. Proposed Methodology

3.1 Research and data collection

During the research for the project, the various research papers were gathered and studied thoroughly and we tried to implement a system that can provide results. During the feature extraction part, just using a basic encoder i.e., CNN and Pooling Layers was not enough, to tackle the issue of losing any important features during the feature extraction phase, we decided to design a custom Encoder model for our need. While finalizing the model design, instead of considering it as a single model, we tried to analyze various parts of the model independently to avoid missing any important factor required in the various parts of the model. Mainly the pre-processing, Feature Extraction, and Image reconstruction. Instead of manually extracting features from the images in the pre-processing part, we created a custom convolutional neural network for encoding i.e., Feature extraction. We tried several methods to train this Autoencoder model consisting of an Encoder and Decoder. We found out that Generative Adversarial Model was the most effective in the training of such model. Thus, it meant introducing a Discriminator model for training the Encoder as well as the Decoder model. The Generator model of the Generative Adversarial network acts as a Decoder in the Autoencoder model. In this way, we were able to efficiently train our Encoder and Decoder models. While doing so, we faced several issues such as diverse image types, varied contexts of images, variable image size, and so on. We addressed many of these issues in further implementing the models. Additional steps such as regularization and normalization of the images were added to overcome some of the challenges. Once we had completed the base models, we developed a web interface as well as a command line interface (CLI) for easy access and usage of the algorithm.

As for the dataset, many datasets were reviewed like Image Dataset and Portrait Paintings. Image Dataset consisted of random pictures and Portrait Paintings had images that were quite similar to each other. But we wanted a dataset with a variety of photos of a particular kind in sufficient numbers so that it can be used for training. Finally, we came across The Stanford Dogs dataset. It consisted of photos of dogs. It included photos of 120 breeds of dogs from all around the world. The dataset had a total of 20,580 images. It had images of a single breed of dog with different backgrounds, different environments, and doing different activities. This variety in the images of a single breed of dogs proved to be helpful in the training process.

3.2 Training Phase

The actual architecture of the model varies according to the phase of the development lifecycle. During the training process, the model acts as a single entity.

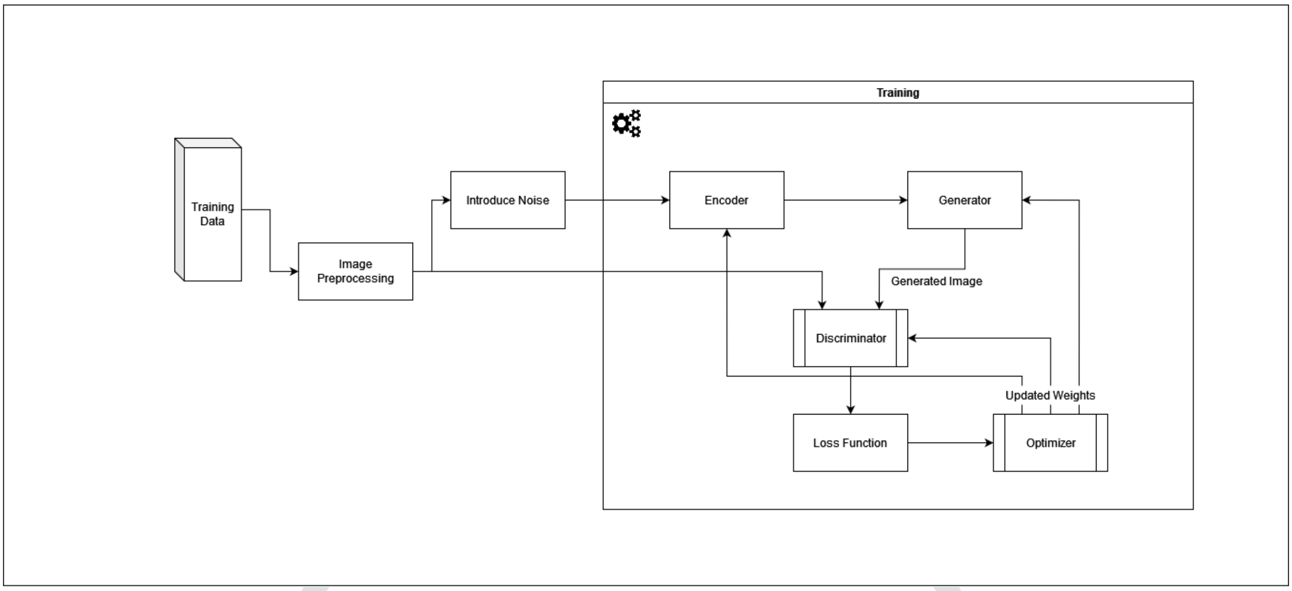

Figure 2 Training model

1. In the Feature Extraction model which acts as an Encoder, the image goes through several convolution layers and pooling layers to extract features from the images. This feature tensor is then passed to the GAN models.
2. Here, the Generator model tries to interpret the features and construct the image.
3. The output image of the Generator is compared with the original image with the help of the Discriminator model, to test the model, and compute the loss.
4. Then the trainable part of the system is altered by the optimizer to reduce the loss.

3.3 Deployment Phase

In the deployment or application phase, the application is divided into 2 parts, the compression model and the decompression model.

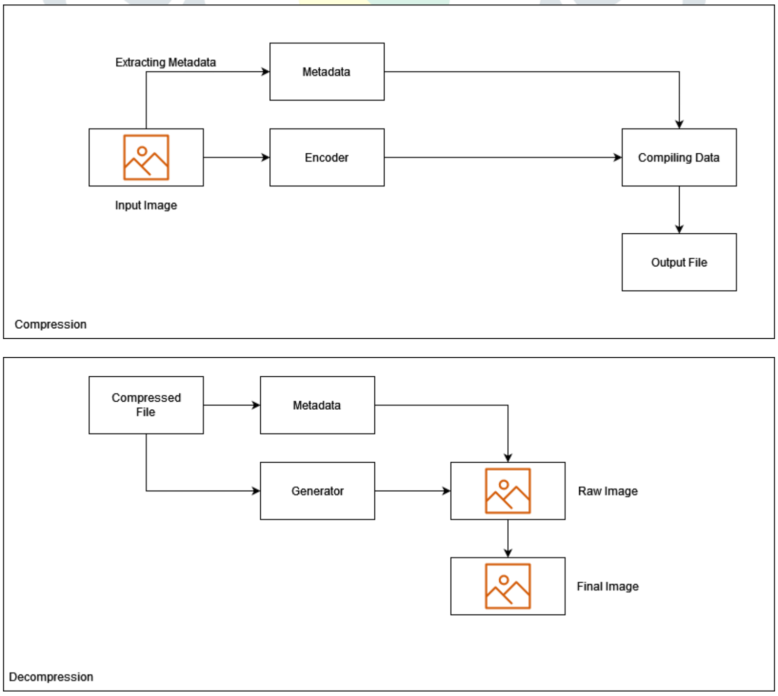

Figure 3 Deployment model

This is very similar to the training phase, but the model is separated during the feature extraction and GAN. The metadata is also handled and preserved during the process and is a quality-of-life feature of the model. Also, the dimension of the image needs to be preserved to construct the image to the specific dimension during the decompression phase. This is easily achievable as both, the Generator model and the Discriminator model are convolutional.

3.4 Encoder Model

The encoder model acts as the compression model, it takes an image as input and converts it into a multidimensional feature array that contains the bare essential features required to reconstruct the image. The layers of the feature matrix can be adjusted to manipulate the compression ratio. This result is achieved using multiple convolutional networks. They dynamically capture various features of images while simultaneously reducing the size of the image.

3.5 Generator Model

Once the features of an image are extracted, i.e., the image is compressed by the encoder, we need to decompress the image, i.e., rebuild and retrieve the original image based upon the encoded information. This task is conducted by the generator model, it takes the encoded image and rebuilds the original image based on the provided features. Thus, after training is completed, the Encoder acts as a compression engine whereas the Generator acts as a decompressor that returns the actual image.

3.6 Discriminator Model

As our AI model is built over the concept of GAN, the discriminator model is used to train the encoder & generator in generating images that are very close if not identical to the original image. To achieve this the model takes the original input image and the output image from the generator and compares them to provide a value expressing how identical the two images are. In the training phase, the loss of the discriminator is calculated as follows:

$$
\begin{aligned}
E_{(out)} &= E(x) \\
G_{(out)} &= G(E_{(out)}) \\
Loss_{(d)} &= \max[\,\log(D(E_{(out)}, x)) + \log(1 - D\big(E_{(out)}, G_{(out)}\big))] \\
& G \rightarrow Generator, E \rightarrow Encoder, D \rightarrow \ Discriminator
\end{aligned}
$$

This loss is then used with Adam optimizer to generate gradients which can later be applied to the trainable variables of the model.

3.7 Assessment Parameters

We computed various parameters, which were used as assessment parameters. These parameters helped us to find out how close we were to the compression of original images. The compression ratio was one of the ratios that we used; it was used to know how much compression we were able to achieve at the end of the compression process. The other parameter we used is SSIM i.e., structural similarity index measure. It is a perceptual metric; it helps us to find the degradation in the original image that has possibly occurred during the process of compression. The final parameter that we used is PSNR i.e., peak signal-to-noise ratio. Usually, the PSNR metric is not used as an assessment parameter in AI-based compression algorithms because a single pixel change during the compression can lead to lower values. But we were able to achieve remarkable structural similarity which enabled us to use PSNR for evaluating our model.

3.8 Web app and CLI for utilizing the model

To provide easy accessibility to the model implementing interfaces was a crucial part of the development process. For CLI (command-line interface) by using the `argparse` module, commands like compress, decompress and compare were implemented. By creating an exportable API for the model, the functionality for compressing and decompressing could be accessed easily. To provide a web interface `fastapi` module was used. To avoid any inconvenience to the user while performing the actions like compression and decompression, background_jobs were used to make it completely asynchronous. Making the website completely responsive in the meantime.

IV. RESULTS

4.1 Assessment parameters and visual inspection

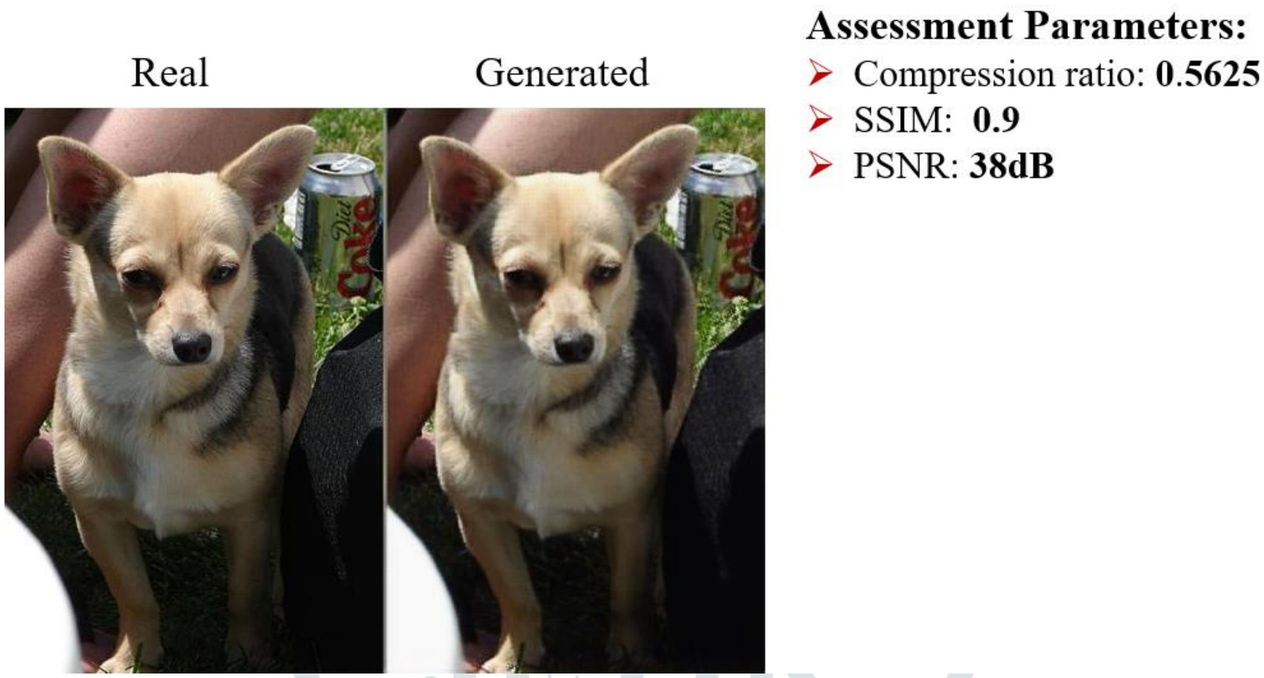

Figure 4 Comparison between real and compressed image

Figure 5 Images generated by the generator

4.2 Web application

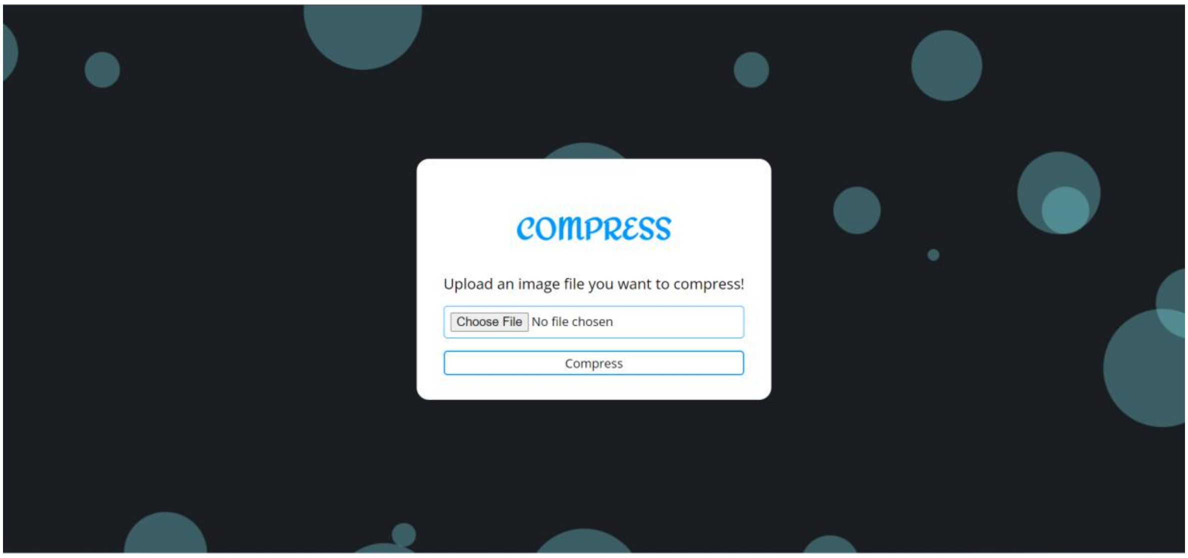

Figure 6 Web interface for demonstration

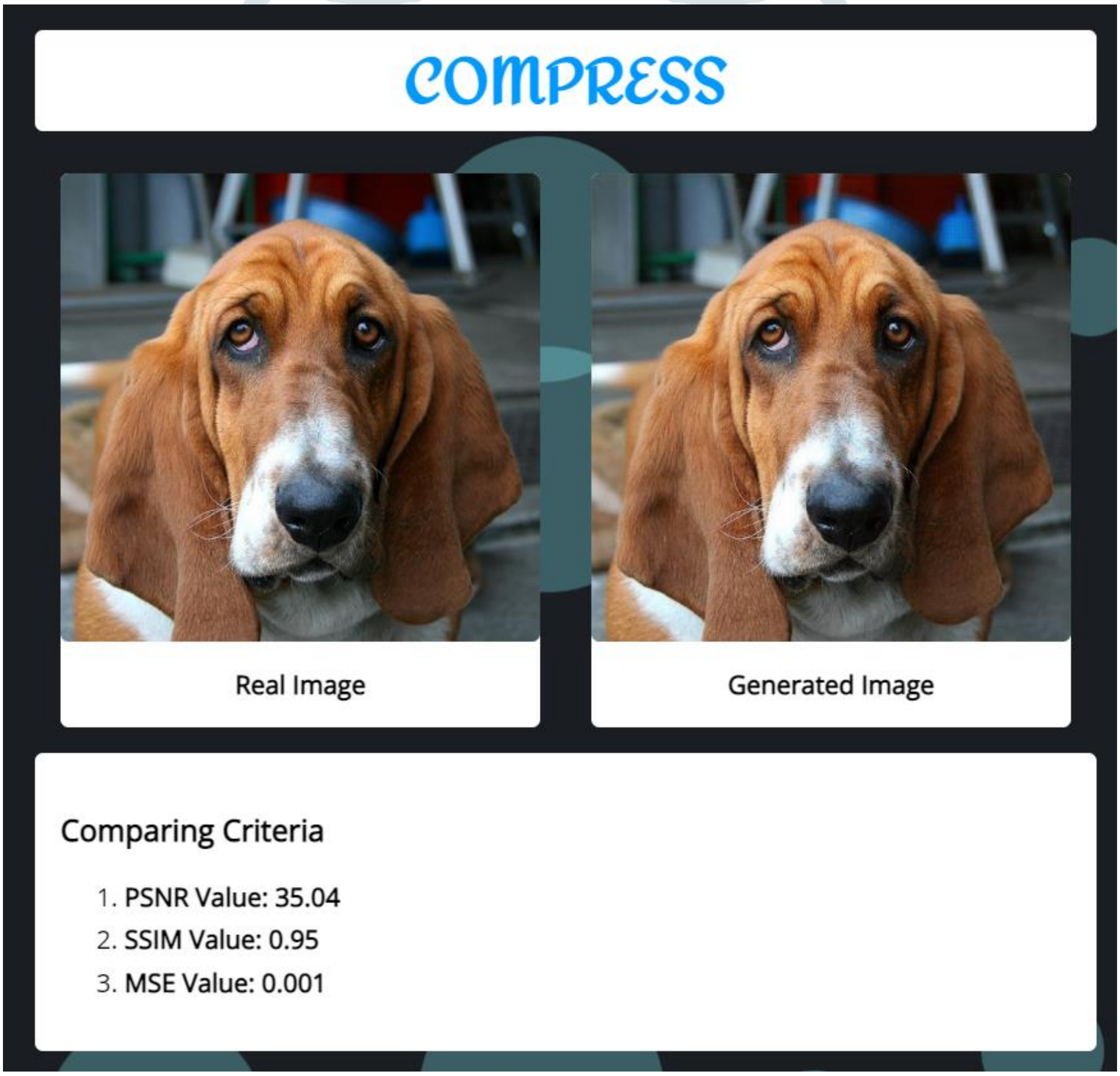

Figure 7 Criteria Comparison in Web Interface

V. Future Scope

A larger dataset can be used with the support of a higher processing power system to increase the accuracy of the existing model. A faster server would reduce the time required to get an output on the client side. Moreover, the web server must be optimized to obtain a faster and more efficient application. Based on this model, a video compression standard can be developed using the same principles. More accurate assessment parameters need to be developed to figure out the actual efficacy of the model as the existing ones provide an incomplete overview of the model.

VI. CONCLUSION

We've successfully implemented and built an Image compression tool using the Generative adversarial model. The Encoder model, that is the feature extraction model was trained using a Generative adversarial network. A dataset of just 3500 images was taken due to the limited processing resources in our possession. Out of which 2500 images were used for training and 1000 image were used for evaluating the model. Even with this limited dataset, the model was able to achieve remarkable results. Using our algorithm, we were able to achieve the compression ratio of 0.5625 while maintaining an SSIM of ~0.9 i.e., ~90% structural similarity. We've developed a CLI application as well as a Web GUI for the demonstration and easy usage of the application. The compression algorithm is optimized for functioning on any device with an active internet connection.

VII. ACKNOWLEDGEMENT

We remain immensely obliged to Prof. Chitra T. Wasnik for providing us with the idea of the topic, for her invaluable support in gathering resources, and for her guidance and supervision which made this work successful. We are also thankful to the faculty and staff of the Computer Engineering Department and Lokmanya Tilak College of Engineering, Navi Mumbai for their invaluable support. We would like to say that it has indeed been a fulfilling experience working out this project topic.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," Advances in Neural Information Processing Systems, pp.2672-2680, Dec.2014.

[2] Shinobu Kudo, Shota Orihashi, Ryuichi Tanida, Atsushi Shimizu, "GAN-based Image Compression Using Mutual Information Maximizing Regularization", 2019 Picture Coding Symposium (PCS), Nov. 12-15, 2019, Ningbo, China.

[3] O. Rippel, and L. Bourdev, "Real-Time Adaptive Image Compression", arXiv preprint arXiv:1705.05823v1, May 2017.

[4] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. V. Gool, "Generative Adversarial Networks for Extreme Learned Image Compression", arXiv preprint arXiv:1804.02958, Apr. 2018.

[5] Pierre Baldi, "Autoencoders, Unsupervised Learning, and Deep Architectures", JMLR: Workshop and Conference Proceedings 27:37–50, 2012

[6] 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.

[7] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.

[8] David Minnen, Johannes Balle, and George D Toderici. "Joint autoregressive and hierarchical priors for learned image compression. In Advances in Neural Information Processing Systems" , pages 10771–10780, 2018.

[9] Oren Rippel and Lubomir Bourdev. "Real-time adaptive image compression. In Proceedings of the International Conference on Machine Learning, Proceedings of Machine Learning Research", pages 2922–2930, International Convention Centre, Sydney, Australia, 2017.