ISSN 2347 - 3983
Volume 8. No. 5, May 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter29852020.pdf
https://doi.org/10.30534/ijeter/2020/29852020

Efficient Image Compression by Machine Learning

K Hema[1], Dr. Mylara Reddy C[2], Rajesh S M[3], Kamala L[4]
[1]Asst. Prof., Dept of CSE, GITAM School of Technology, Bengaluru, India, hkrishna@gitam.edu
[2]Asst. Prof., Dept of CSE, GITAM School of Technology, Bengaluru, India, mchinnai@gitam.edu
[3]Asst. Prof., Dept of CSE, GITAM School of Technology, Bengaluru, India, rshivaga@gitam.edu
[4]Asst. Prof., Dept of CSE, GITAM School of Technology, Bengaluru, India, klakshmi@gitam.edu

ABSTRACT

Media broadcasting with video streaming is booming day by day, image data compression is becoming essential. The key goal of image compression is to obtain a relatively small bit rate and a good visual quality of decompressed images. Artificial intelligence and machine learning are such technologies to carry out this task. Each picture is composed up of pixels. Those pictures are labeled as noisy pictures. We propose a convolution auto encoder neural network to compact the images by using the MNIST database where we up sample and down sample an image to increase and decrease its quality. We take 128 by 128 dimensional image. By generating a deep learning machine image, the image must be compressed and transformed to 128 by 1 dimensional vectors in order to obtain a clear original picture type. The main objective is to compress the picture without reducing its quality, predicting the value present in the compressed picture of the MNIST database.

Key words: Artificial intelligence, deep learning, MNIST database, machine learning.

1. INTRODUCTION

Machine Learning (ML) is one of the AI's typical functions in which machines, algorithms, and devices function via intelligence. Photo processing is used to remove photographs' color and texture characteristics such as red, green, gray, fade, saturation, density, entropy, strength, contrast, homogeneity [6]. Machine Learning essentially means that we are guiding the system to perform something (here image processing) by furnishing a set of guiding inputs. Machine Learning have portraits, failure operations and certain methods that can be pre owned to resolve which would provide better image processing. Also, it confides on the class of image processing we aim to perform as there are certain failure operations that operate well than other due to their built in assets. For example there is high possibility that the cross-entropy failure operation could do well than other failure operations to provide a preferred image processing.

1.1 Classification of Compression Algorithms

Compression is a mechanism which takes an input D and produces C (D) with the lesser number of bits in comparison to
input. The backward mechanism termed as decompression accepts C (D) and regenerates the data D as presented in Figure 1.

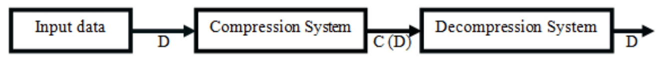


Figure 1: Compression Algorithms

Compression can split into two groups, as lossless and lossy compression [1], and is shown in Figure 2.

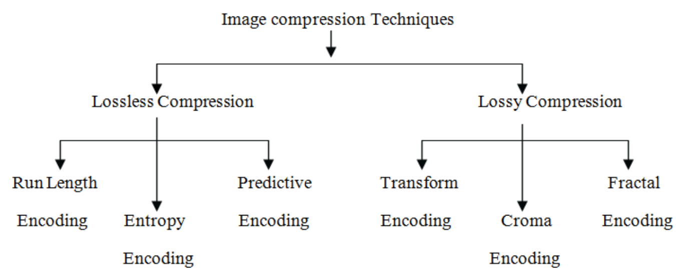


Figure 2: Image Compression Techniques

Compression transforms picture information for approximations in the simplest sense. These modifications are imperceptible to the human eye at low compressive rates. As the compression degree rises, the adjustments can become more visible. Compression matters to the quality of the picture, just in the same way as image resolution is relevant. A high-resolution, extremely compressed image can look terrible. An image with low resolution and no compression will look terrible. But a high-resolution picture with a fair amount of compression will look as fine as a high-resolution picture without any compression. A text file or application may be compressed without errors but only to some degree. This is called loss less compression. Beyond this point, errors are introduced. It is important in text and computer files to loss less compression because a single error will significantly harm the context of a text file or trigger a computer not to run. A slight reduction of efficiency is typically not apparent of image compression.

There is no "trigger level" at which compression operates well, however it is difficult after this. When there is any resistance to failure, the compression effect can be higher than it is where there is no resistance for losses. For this reason visual images should be more compact than text files or applications. The aim of lossless image compression is to impersonate an image sign with the fewest possible available number of bits without damaging any data, there by fasting up transmission and decreasing stockpile demand.

1.2 Advantages of Image Compression

A. Size contraction:

The size contraction is the biggest gain in image compression. It occupies a limited area on the hard disk and maintains the similar environmental size, until we modify the picture's environmental size in a picture rewriter. Reducing the size of the file would build picture rich sites without much bandwidth division or stockpile capacity.

B. Loss of Data:

In files such as JPEG, an image decreases in the scale of the compression, can delete any of the photo details forever. So shorten the files to assure that the backup exists before beginning. Else the high quality of the initial decompressed picture will be lost forever.

C. Sluggish Gadgets:

Diverse computerized gadgets loads hugely compressed pictures gently. For instance Compact Disk gadgets can only capture information at a definite pace and cannot exhibit broad pictures in real time. Image compression grants the rapid gathering of information on sluggish gadgets.

1.3 Motivation

Human beings are extremely visual creatures. Evolution has invested a significant proportion of our neural capital into sensory vision. We are specialists at catching visual objects in a fraction of second and focus on visual knowledge for much of our everyday activities. It is also not shocking that as our planet is more digital every day, digital photographs and digital video are becoming more accessible and widespread. Therefore, it is necessary to measure the accuracy of a picture and to compress the picture as the capacity available is minimal. There are quantitative tests to test the picture quality, but they have certain disadvantages. To solve these inconveniences we need a model that can compact the picture while retaining its accuracy.

2. RELATED WORK

Huffman Coding is a lossless encoding strategy that provides codes based on the recurrence of the pixels. Pixels recurring very often will have smaller codes whereas pixels recurring less often will have the same length codes. The outcome is variable length codes that contain an integral number of bits. The codes end up having unique prefix property. The codes are stored on a binary tree. The binary tree is built using the codes, starting from the leaves and moving up to the root of the tree. An experiment to compact a gray scale picture using Huffman coding algorithm was conducted. Huffman coding has been used to decrease the amount of bits required to imitate each pixel. Experiment results showed that up to a 0.8456 compression ratio was obtained on the image [2].

Deep Auto encoder's were presented where the first step was to train the Auto encoder. Here the pictures were first passed into the nonlinear encoding slab. This encoding slab gradually compact's the input image pixels into a compact vector. After the encoding was complete, the compact vector was again passed into the decoding layer, and then reversed the encoding layer cycle to achieve input picture reconstruction. The deviation across the reconstructed image and input picture was assumed as error to help decrease the gap. Also, a sparsity parameter was used to ensure sparsity of the activation vectors in each layer except the output layer. The next step was to discover an excellent group of quantization values using k-means clustering to encode activation values. Finally, they compared this compression algorithm with other familiar techniques like JPEG and PCA. They observed that Deep Auto encoder image compression technique outperformed JPEG in huge compaction and decreased resolution system. Deep Auto encoder also demonstrated improved non-linear representation of the input picture than PCA's and therefore Deep Auto encoder had higher accuracy of reconstruction. They found that Deep Auto encoder was able to evaluate statistical regularities in a specific domain of pictures which JPEG could not do [3].

An Auto encoder was presented for diminishing the dimensionality of information. They trained a bundle of RBM's for neural network pre training with every RBM consisting of a layer of functionality detectors. After the pre coaching, the Restricted Boltzmann machine were uncovered to grab full slabs of the neural network which at last were fine-tuned using back propagation algorithm to reduce the data reconstruction error. After pre coaching the network, the encoder part of the neural network is then unrolled to get the decoder part where similar weights of Auto encoder are used. The weights are later adjusted in the fine tuning stage to get optimal reconstruction. They in general conclude that back propagation together with Deep Auto encoder is a brute force for effective representation and generalization of nonlinear structure data [4].

Large-scale unsupervised learning algorithms were used to create high-level characteristics that require the use of RBM stacks to construct Auto encoder's. Training was given to produce high-quality; class-specific trait discoverer's using unlabeled data. They have trained a nine layer locally bound scant automatic encoder with local receptive fields, pooling and local contrast normalization on a huge database of pictures. They proved that it is feasible to guide a face discoverer without label pictures having a face. The idea behind this approach is that only a specific area of the lower layer can be linked to each feature in the auto encoder. They have employed local L2 pooling and spatial contrast normalization to gain invariance to local deformations. The deep auto encoder was created by doubling 3 times the equivalent stage consisting of local contrast normalization, local filtering and local pooling. There is no sharing of weights that grants study of in variances other than translational variances. In short they demonstrate that Deep Auto encoders work well as feature detectors for images [5].

3. SYSTEM DESIGN

3.1 System Architecture

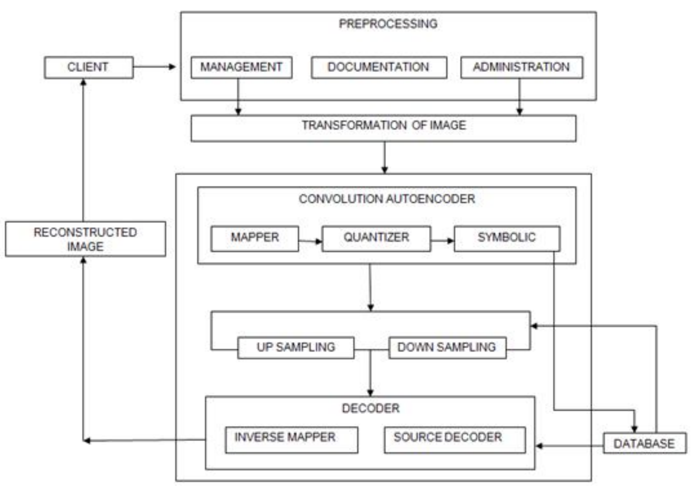


Figure 3: System Architecture

Figure 3 shows the architecture of the system and is explained below.

A. The Client:

An individual or an organization is the client using a

Skilled employee or a company's resource. The client sends the picture that has to be compressed here (As the picture is from the MNIST database, it is a manuscript digit dataset with a 28* 28pixel image size).

B. Pre-processing:

Picture pre-processing is similar to the mathematical normalization of data collection, which is a typical stage in compression of the picture, photo resizing.

C. The Transformation of image:

When transforming, most styles of layers in the image, such as pixel layers, sort layers, bitmaps, agile items mounted are quietly dragged at an edge knob to proportionally scale certain styles of layers. This transforms the image to a complex vector of 128 dimensions.

D. Convolution Auto encoder:

To boost the efficiency we might attach more layers to render the network wider. Nevertheless, since we're operating on images, we might use convolution neural network to boost compression and decompression accuracy.

E. Auto encoder:

There are two components of an auto encoder: an encoder and a decoder. The encoder restricts the dimensions of the input picture to compact the initial picture. The decoder recovers the original picture from the compressed picture. Auto encoder may be used to compact pictures. In this task, the size of the hidden layer in the auto encoder is strictly smaller than the output layer size. Training these auto-encoders with the input values exactly same as the goal values causes auto-encoders to know the low-dimensional representation of the input data. The Hidden layer is enabled by the compressed data. The preceding portion of this network is the picture encoder and the last component is viewed as the decoder [3]. Some auto encoder's will reduce the MNIST database numbers from 784 pixels to 32 pixels without much loss from input to output, like the code on the Keras auto encoder.

F. Sampling:

There are two forms of it: Up sampling and down sampling.

Up sampling an image means improving the accuracy of the pictures.

Down sampling is the method of reducing the size of a picture without losing the accuracy.

G. The Database:

A database is a structured data set, typically maintained and electronically accessible by a computer network.

H. Reconstruction of an Image:

This recreates the compressed image that has 128 dimensional matrixes; it is recreated to an initial image form. It sends the compressed image to the client.

3.2 Event diagram

The events in the application are shown in Figure 4. First of all, the network is prepared for 10000 iterations. In this step the network knows to compact and recreate the initial picture as accurately as possible. The performance of the network is then checked. Secondly, the user uploads a picture to the application. The application then compresses the picture, returning the compressed picture. In the final stage, the consumer uploads an already compressed picture which is decompressed by the application and returns the reconstructed initial picture.

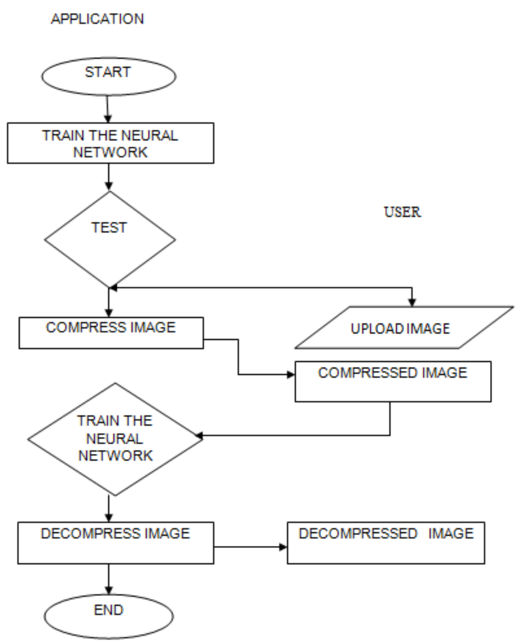


Figure 4: Event diagram

3.3 Methodology

A. Data Collection:

The handwritten numeral pictures from the MNIST archive are used for data collection for training and testing. The pictures are of size 28x 28 (784 total pixels in the picture) and thus the input layer size is 784. The MNIST set includes sixty thousand pictures in total. First “20000 pictures” are utilized, for training and “1000 pictures” are utilized for testing. The sample data is shown in Figure 5.

255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	248	216	216	216	56	47	20	200	25	0	1	55
255	255	255	255	255	192	181	91	33	23	0	0	0	0	0	3	22	0	1	12	133
255	255	255	255	157	1	0	0	0	0	0	0	0	0	0	92	107	107	146	173	255
255	255	255	255	216	4	0	0	0	0	0	10	17	1	1	255	255	255	255	255	255
255	255	255	255	255	109	32	76	0	0	8	231	255	168	33	255	255	255	255	255	255
255	255	255	255	255	255	224	253	33	0	96	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	45	0	13	250	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	231	13	0	124	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	183	1	3	29	74	253	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	108	1	0	0	63	202	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	164	15	0	0	36	198	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	220	92	0	0	15	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	0	0	0	133	255	255	255
255	255	255	255	255	255	255	255	255	255	255	163	52	17	0	0	8	250	255	255	255
255	255	255	255	255	255	255	255	255	175	37	3	0	0	0	0	17	255	255	255	255
255	255	255	255	255	255	255	204	68	4	0	0	0	0	9	112	255	255	255	255	255
255	255	255	255	255	206	130	6	0	0	0	0	10	108	250	255	255	255	255	255	255
255	255	255	216	23	4	0	0	0	0	12	109	235	255	255	255	255	255	255	255	255
255	147	22	3	0	0	0	0	1	50	231	255	255	255	255	255	255	255	255	255	255
255	47	0	0	0	6	48	51	220	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

Figure 5: Dataset showing digit 5

B. Use of CNN:

CNN is a most powerful supervised deep neural learning technique. In addition, the eventual configuration of the CNN is significantly close to standard Nets, where neurons have weights and prejudices. We also adopt a loss function (ex: cross entropy, soft max) and an optimizer (e.g. Adam optimizer) in Convolution neural networks, much as we do in standard Nets. Furthermore, there exists the Convolution Layers, Pooling Layers and Flatten Layers in CNNs. CNNs, primarily used for image recognition, although other applications such as, natural language processing can be found[1]. Regular Nets 'key fundamental function is that all the neurons are interconnected. For example, if we have 28 by 28 pixel images with just grayscale, we would finally result with seven hundred and eighty four (28 x 28 x 1) neurons in a slab that might be controllable. Some pictures therefore contain more pixels and are not grey-scaled. Therefore, if we bring a series of color pictures in 4 K Ultra High Resolution, we will have 26,542,080 (4096 x 2160 x 3) different neurons in the first layer attached to every individual which is not quite simple to handle. Therefore we may assume that Regular Nets cannot be scaled for image classification. Nonetheless, there appears to be limited connection or interaction between two independent pixels, particularly when it comes to pictures, except closer to one other. It refers to the concept of Convolution Layers and Pooling Layers.

C. CNN Slabs:

In the convolution neural network, we are able to use several specific slabs. And the most significant levels are convolution, pooling, and fully connected layers.

- The Convolution Layer

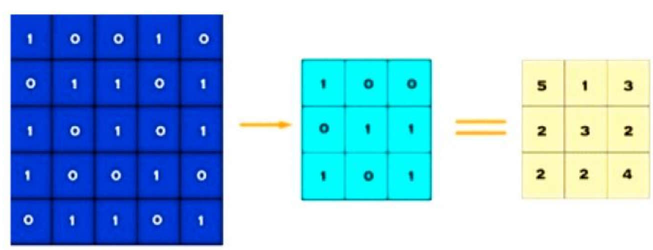


Figure 6: The Convolution Layer

Convolution layer as shown in Figure 6 is the foremost layer from which the characteristics of the pictures are derived in to our database. Convolution comprises overlaying and determining a weighted total of all pixel values [7].Convolution helps us in maintaining an association between various sections of a picture assuming that pixels are always connected to nearby and neighboring pixels. Convolution effectively filters the picture with lesser pixel filter to minimize the picture capacity, without sacrificing the connection among the pixels. A three by three filter with one by one phase can be used to apply convolution on 5x5 images, we end up with a 3x3 output (with a complexity decrease of 64 per cent).

- Pooling Layer

While constructing CNNs, pooling layers are typically added to diminish the dimensional capacity of depiction to scale down the number of parameters that decreases the calculation intricacy. Additionally, pooling layers often assist the issue of over fitting. Generally, pooling size is chosen to minimize the number of parameters by choosing the limit, total values or average inside such pixels. One of the popular pooling methods, Max Pooling is shown in the below Figure 7.

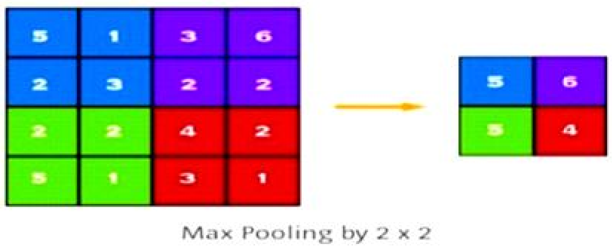


Figure 7: Max pooling

- Fully Connected Layers

Our Regular Net is a completely connected network in which each parameter is connected together to decide the correct relationship and impact of every parameter on the labels. Thanks to convolution and pooling layers, as time-space complexity is considerably reduced, a fully connected network can be constructed at last to segregate the pictures. The Collection of layers that are completely linked looks as shown in the Figure8 and sample convolution neural network is shown in Figure 9.

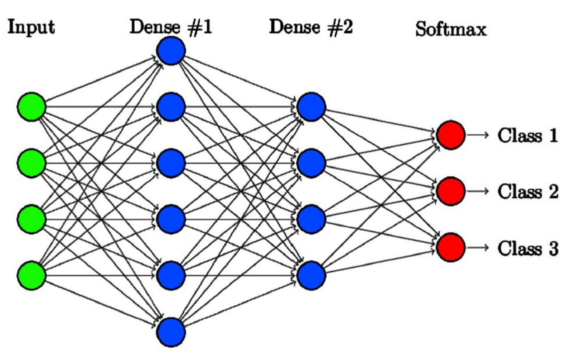


Figure 8: A fully connected neural network

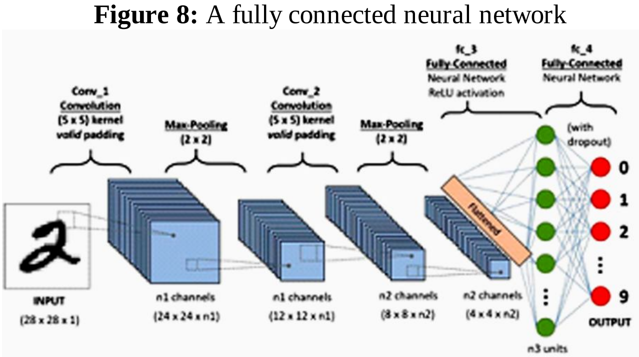


Figure 9: Example of Convolution Neural Network

4. EXPERIMENTAL SETUP AND RESULT

A. Downloading the MNIST Data:

The MNIST dataset is a popular picture segregation database and is available from several distinct outlets. Indeed Tensor flow and Keras confess us directly to get and load the MNIST dataset from their Application Program Interface. The MNIST database archive includes sixty thousand practicing pictures and ten thousand trial pictures provided by ACB employees and students in the USA. Furthermore, we split the 2 classes as train and test and also divided the labels and the pictures. The sections x_train and x_test includes grey scale Red Green Blue codes ranging 0 to 255, while the sections y_train and y_test bear labels from 0 to 9 showing the numeral they really are. We can get support from matplotlib to visualize such numbers.

Grey scale visualization image appears as shown in the Figure 10.

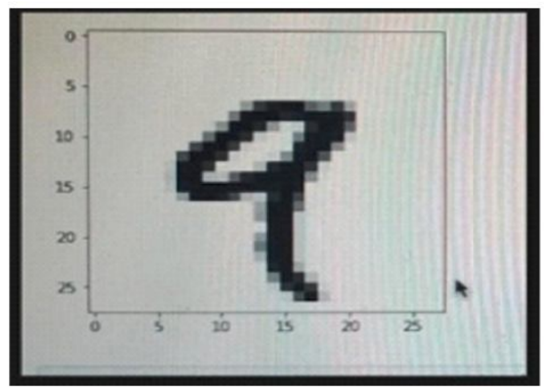


Figure 10: Visualization of sample image

B. Training:

We will need to learn the dataset structure in order to funnel it through the convolution neural network. So we can use numpy array's "shape" attribute. We are going to have this (60000, 28, 28). As we expected, 60,000 is the amount of pictures in the practicing dataset and (28, 28) is the dimension of the picture twenty eight by twenty eight pixels.

C. Transfiguring and Standardizing the Pictures:

We require 4-dims numpy arrays to use the database within Keras Application Program Interface. However our list is 3-dims, as we saw above. We do need to standardize our results, as it is often important in neural designs. It may be achieved by splitting the Red Green Blue codes to 255 .

D. Constructing the Convolution Neural Network

A model with high-level Keras APIs is developed that can use Tensor Flow in the rare end. We also indicate that there exists many high level Tensor Flow Application Program Interfaces, such as Estimators, neural network library and Layers that benefit in creating a high level knowledge neural networks. This will also contribute to ambiguity because they differ in the form of execution. For the same neural network, different color codes can be observed they all use tensor flow.

We are going to use the shortest Application Program Interface that is Keras. Then, we would import the Keras Sequential Template and include Conv2D; the Max Pooling and Flattening.Epoch number might seem a little low. We can therefore hit a check accuracy of 98–99 percent. Since the MNIST dataset doesn't need strong computational power, you can also easily explore by using larger epoch value.The summary of buliding CNN is shown in Figure11.

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 28, 28, 1)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 28, 28, 16)        160
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 14, 14, 8)         1160
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 7, 7, 8)           0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 7, 7, 8)           584
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 4, 4, 8)           0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 4, 4, 8)           584
_________________________________________________________________
up_sampling2d_1 (UpSampling2 (None, 8, 8, 8)           0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 8, 8, 8)           584
_________________________________________________________________
up_sampling2d_2 (UpSampling2 (None, 16, 16, 8)         0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 14, 14, 16)        1168
_________________________________________________________________
up_sampling2d_3 (UpSampling2 (None, 28, 28, 16)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 28, 28, 1)         145
=================================================================
Total params: 4,385
Trainable params: 4,385
```

Figure 11: Summary

E. Checking the model:

Eventually, we are evaluating the qualified model by the test data and the results are pretty good for different epoch values. The MNIST archive includes sixty thousand training patterns and ten thousand validation patterns afforded by ACB and school students in the USA.

F. Results:

The train set for this evaluation experiment image compression is randomly selected from the MNIST database. Anaconda Navigator and Jupyter Note-book software platform is use to perform the experiment. The obtained results are shown in Figure12.

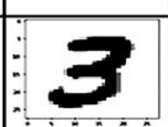
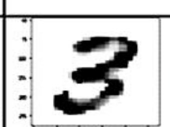
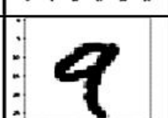
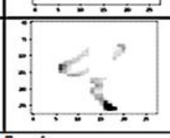
Train Data	Epoch Value	Original Image	Compressed Image	Loss	Predicts the compressed image
X_train[7]	1/1			0.2186	yes
X_train[4]	1/1			0.2186	yes
X_train[7]	1/2			0.2180	yes
X_train[4]	2/2			0.1587	yes

Figure 12: Results

5. CONCLUSION AND FUTURE ENHANCEMENT

This application is able to compress the MNIST handwritten numeral pictures up to 128:1 compression ratio. It also predicts the numeral appearance in the compressed image of MNIST dataset using convolution auto encoder. In the future, this model can be used in hand written data recognition as a real world application.

REFERENCES

1. Amanpreet Kaur, Dr. Jagroop Singh .**Review On Image Compression Techniques And Advantages Of Image Compression,** ISSN: 2319-8354 vol.No.5Issue No.08, August 2016.
2. Nehal Markandeya, Prof.Dr.Sonali Patil. **Image Compression Using Huffman Coding**, International Journal Of Engineering And Computer Science ISSN: 2319-7242 Volume 6 Issue 1 Jan. 2017, Page No. 19999-20002.
3. Anand Atreya and Daniel O'Shea. **Novel Lossy Compression Algorithms with Stacked Autoencoders,** 11 December 2009.
4. G. E. Hinton* and R. R. Salakhutdinov. **Reducing the Dimensionality of Data with Neural Networks.**
5. Quoc V. Le ,Marc'Aurelio Ranzato ,Rajat Monga ,Matthieu Devin ,Kai Chen ,Greg S. Corrado ,Jeff Dean ,Andrew Y. Ng.**High-level Features Using Large Scale Unsupervised Learning** .
6. Jesusimo.L Dioses Jr. **Discrimination of Civet Coffee Using Image Processing and Machine Learning**, Volume 8. No. 4, April 2020, International journal of Emerging Trends in Engineering Research. https://doi.org/10.30534/ijeter/2020/19842020
7. M.V.D Prasad , Syed Inthiyaz ,M. Teja kiran kumar , K.H.S.Sharma,M. Gopi Manohar,Rupa Kumari, Sk Hasane Ahammad. **Human activity recognition using Deep Learning**, Volume 7, No. 11 November 2019. International journal of Emerging Trends in Engineering Research. https://doi.org/10.30534/ijeter/2019/227112019