Image Compression using Convolutional Autoencoders

Team members:

1. Shriya Baskaran (149)
2. Sowmya R (159)
3. Swetha P (182)
4. Swetha Saseendran (183)

Problem Statement

The most basic problem faced by everyone in the 21st century is a time and space(memory) manageable. Digitizing the image consumes more bandwidth which means more cost for the transmission of the data.

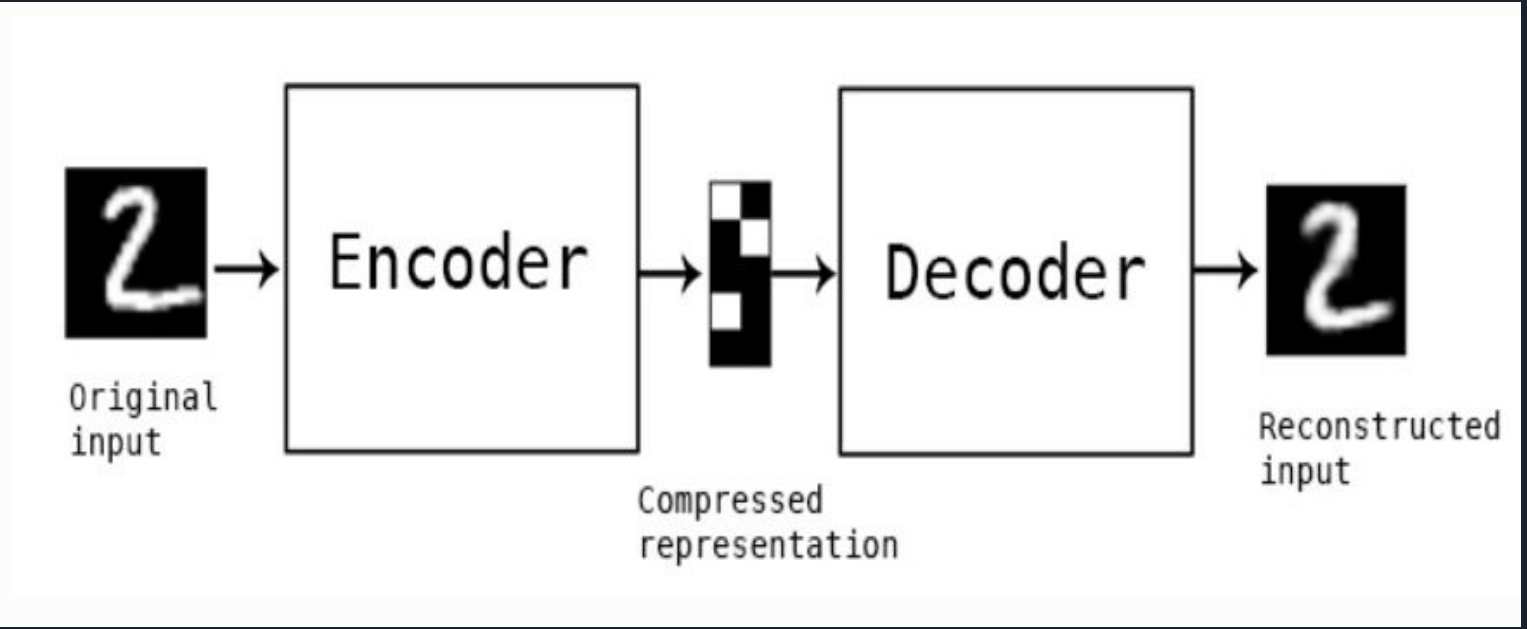
So for the efficient use of bandwidth we use neural networks. Since Image compression is used for faster transmission in-order to provide better services to the user (society). This technology is designed to reduce the resolution of the original image using Convolutional Auto encoder.

The original image of size (let’s say 28 * 28) is reduced to 28 * 1 using the encoder and so the image is compressed after the input to the neural network data points. The main objective is to compress an image without affecting the quality of image radically

Need for compression

- Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level.
- The reduction in file size allows more images to be stored in a given amount of disk or memory space.
- Transmission of files to another device is made much easier.
- Communication cost much lesser when compression is used.
- Communication bandwidth is saved without creating much impact on the image quality.

Methodology



Autoencoders

Autoencoder is an **unsupervised artificial neural network** that learns how to efficiently compress and encode data then learns how to reconstruct the data back from the reduced encoded representation to a representation that is as close to the original input as possible.

Autoencoder is one of the simplest **generative** models. The output of a generative model is an image of the same size and similar appearance as the input image.

- **Data specific** : they will only be able to compress data similar to what they have been trained on.
- **Lossy** : the decompressed outputs will be degraded compared to the original inputs
- **Learns automatically from examples**

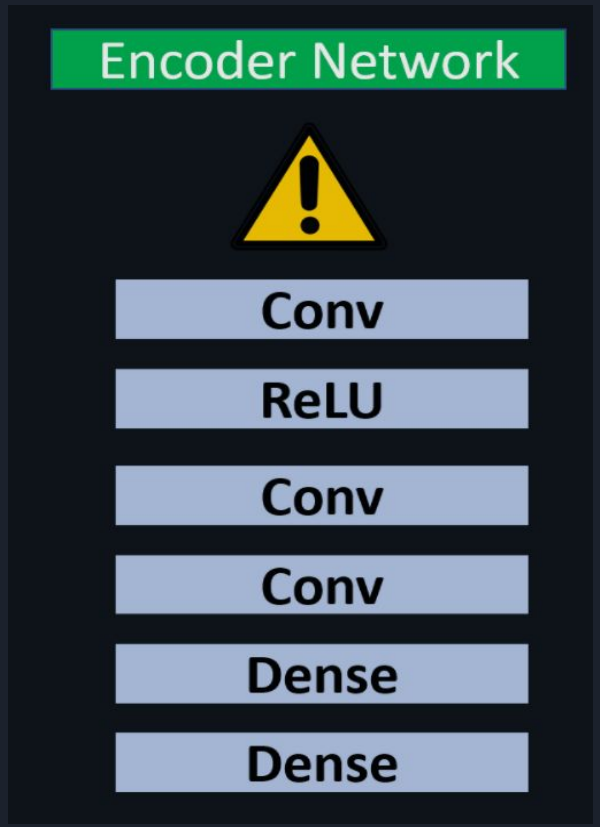
Working

To accomplish this task an autoencoder uses two different types of networks.

- **Encoder** : accept the original data (e.g. an image) that could have two or more dimensions and generate a single 1-D vector that represents the entire image.
- **Decoder** : 1-D vector generated by the encoder from its last layer is then fed to the decoder. The job of the decoder is to reconstruct the original image with the highest possible quality. The decoder is just a reflection of the encoder.

The loss is calculated by comparing the original and reconstructed images, i.e. by calculating the difference between the pixels in the 2 images.

Architecture



Overall Flow of the project

Fetching Dataset.

Applying the Convolutional Filters

Encoding the **Image**

Compressing the **Image**

Decoding the **Image**

Reconstruction and visualization of compressed **Image**

Analysing the Loss and Noise in the Reconstructed **Image**

Deploying the model

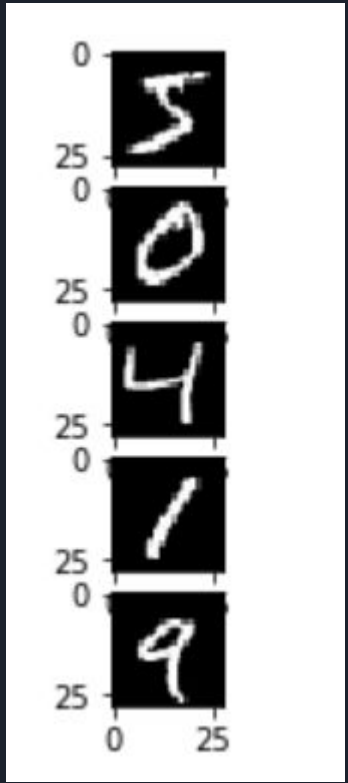
Dataset

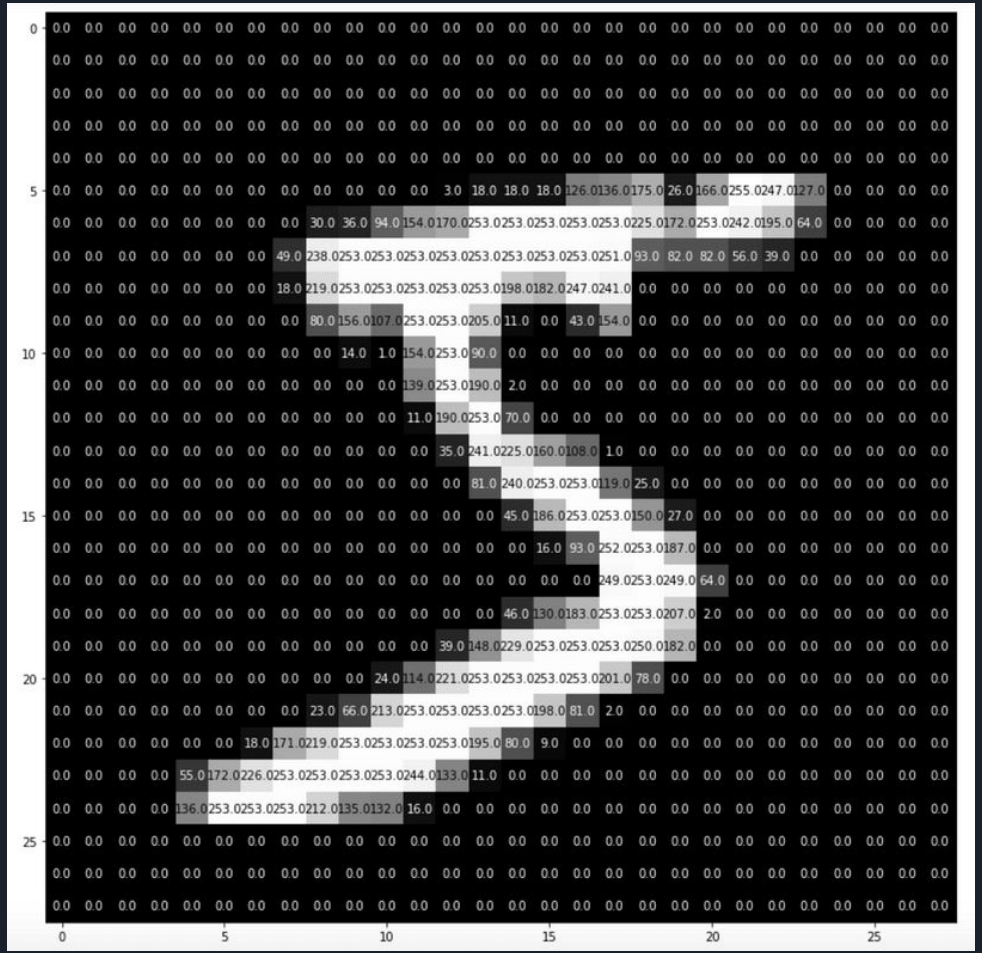
- Gray-scale images of hand-drawn digits, from zero through nine.
- Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total.
- The first column, called "label", is the digit that was drawn by the user. The rest of the columns contain the pixel-values of the associated image.
- The size of each image in the MNIST dataset (which we'll use in this project) is 28x28. That is, each image has 784 elements. If each image is compressed so that it is represented using just 28x1 which is 28 elements and thus :

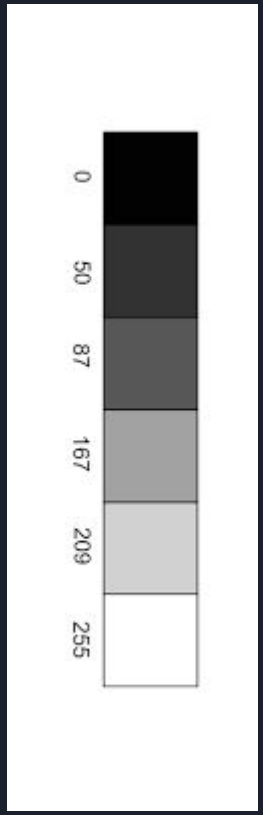
(756/784)*100=96.42% of the data is compressed!

No. of training dataset images = 60,000

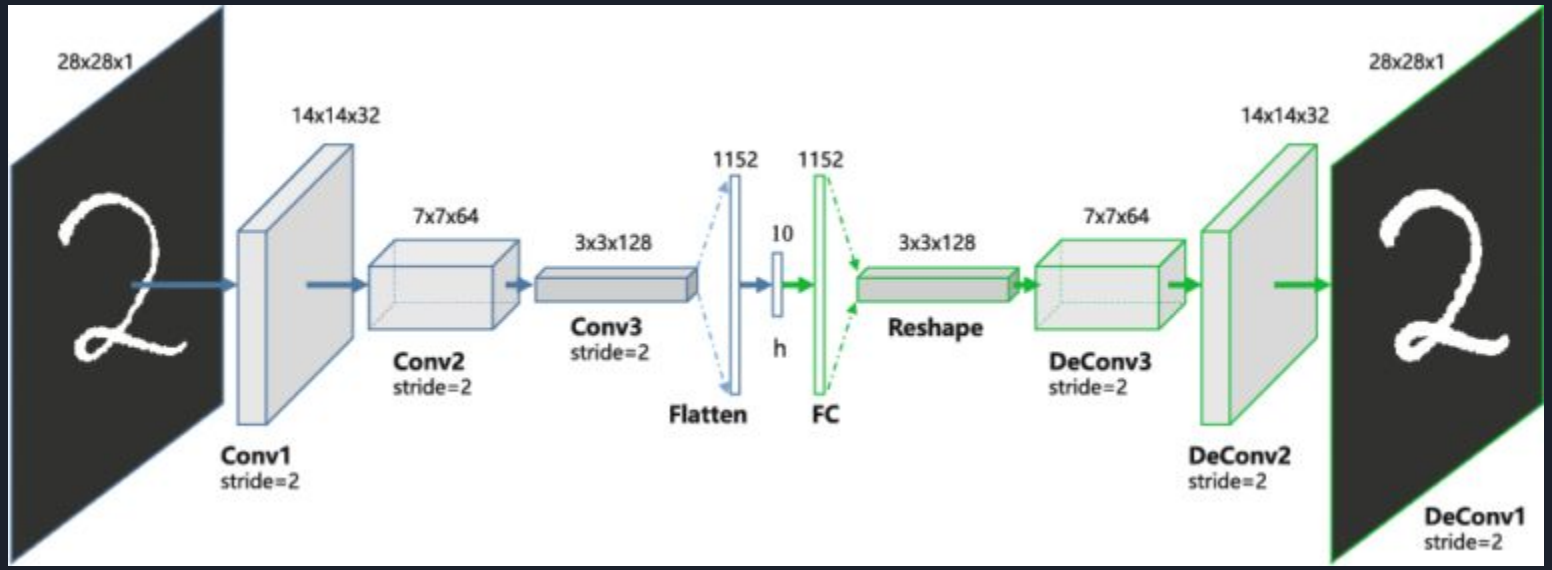
No. of testing dataset images = 10,000



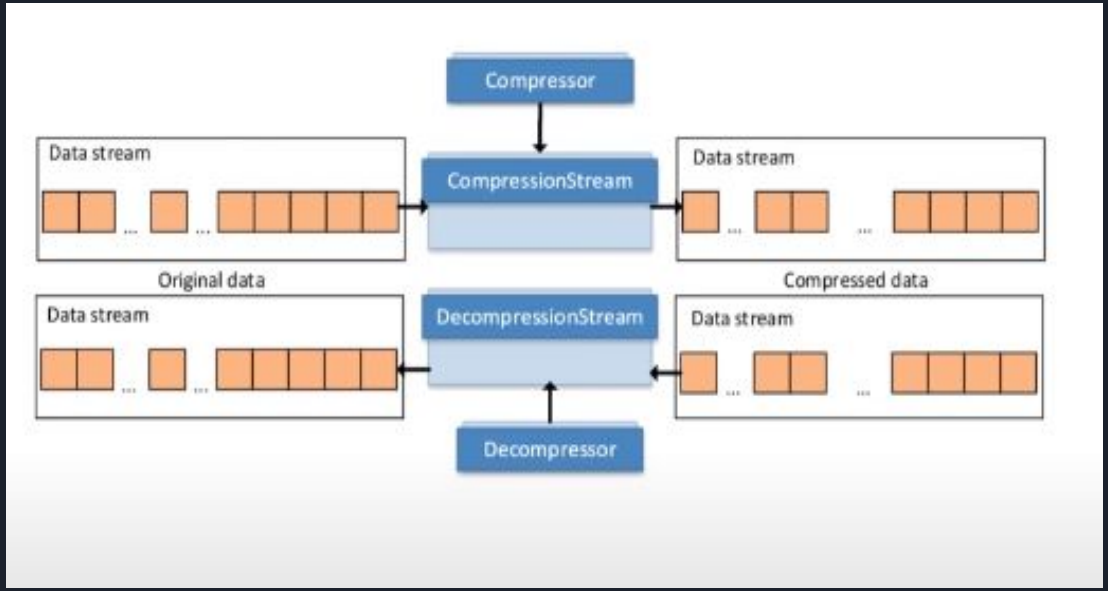

0
5
10
15
20
25
0
5
10
15
20
25


0
50
87
167
209
255

Appling the Convolutional Filters



Encoding and Decoding the **Image**



Analysing the Loss and Noise in the Reconstructed **Image**

LOSS FUNCTION = RECONSTRUCTION LOSS + REGULARIZER

VISUALLY LOSSLESS THRESHOLD (VLT)

PEAK SIGNAL TO NOISE RATIO (PSNR)

THANK YOU