VAE Model Architecture :

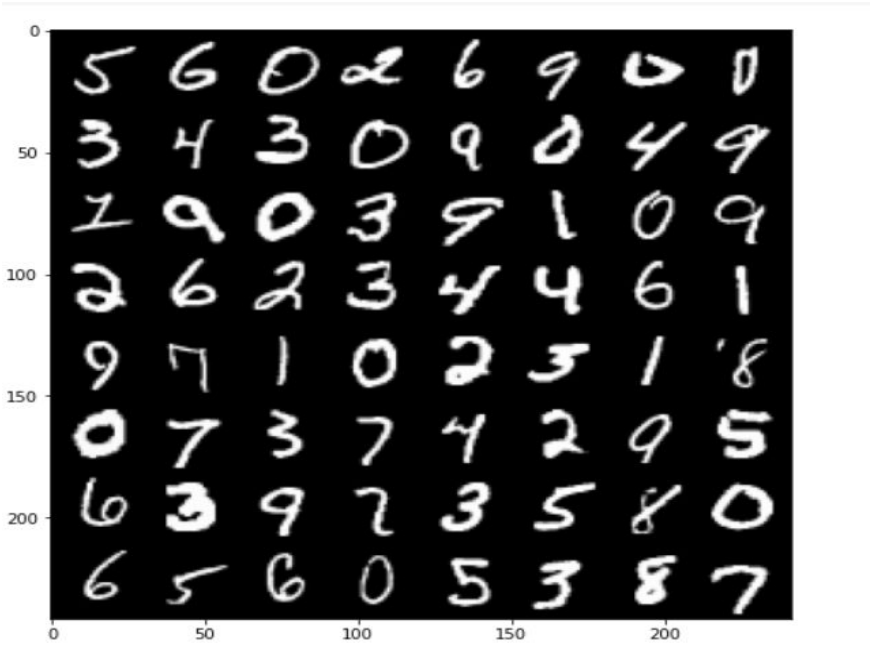
Dimensions of **Latent Data** considered is : 10

Used Convolutional Based Encoder and Decoder for implementing Variational Auto Encoder.

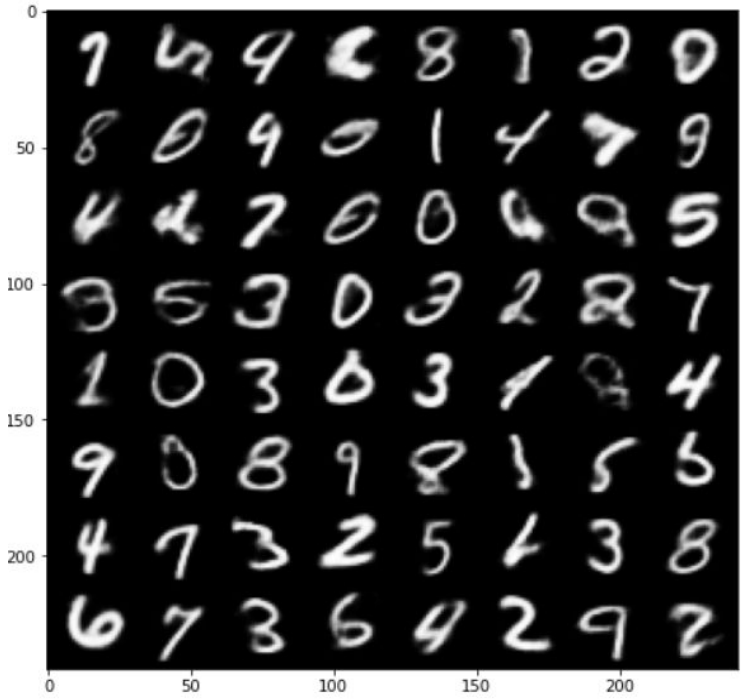
The Architecture used is :

```
VAE(
  (enc): Encoder(
    (conv1): Conv2d(1, 16, kernel_size=(3, 3), stride=(1, 1))
    (conv2): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1))
    (linear): Linear(in_features=18432, out_features=512, bias=True)
    (linear2): Linear(in_features=512, out_features=256, bias=True)
    (mu): Linear(in_features=256, out_features=10, bias=True)
    (var): Linear(in_features=256, out_features=10, bias=True)
  )
  (dec): Decoder(
    (linear): Linear(in_features=10, out_features=256, bias=True)
    (linear2): Linear(in_features=256, out_features=512, bias=True)
    (linear3): Linear(in_features=512, out_features=18432, bias=True)
    (deconv1): ConvTranspose2d(32, 16, kernel_size=(3, 3), stride=(1, 1))
    (deconv2): ConvTranspose2d(16, 1, kernel_size=(3, 3), stride=(1, 1))
  )
)
```

Visualization of MNIST Dataset (Original data) :



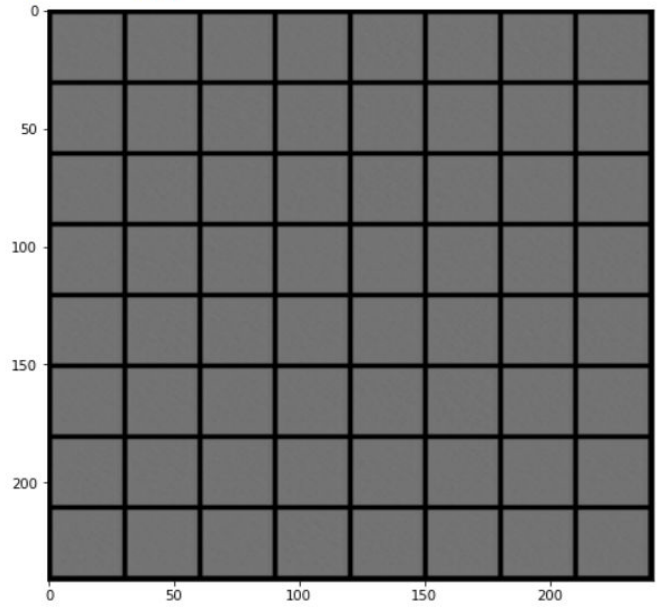
Visualization of Generated Data by using VAE : (after training)



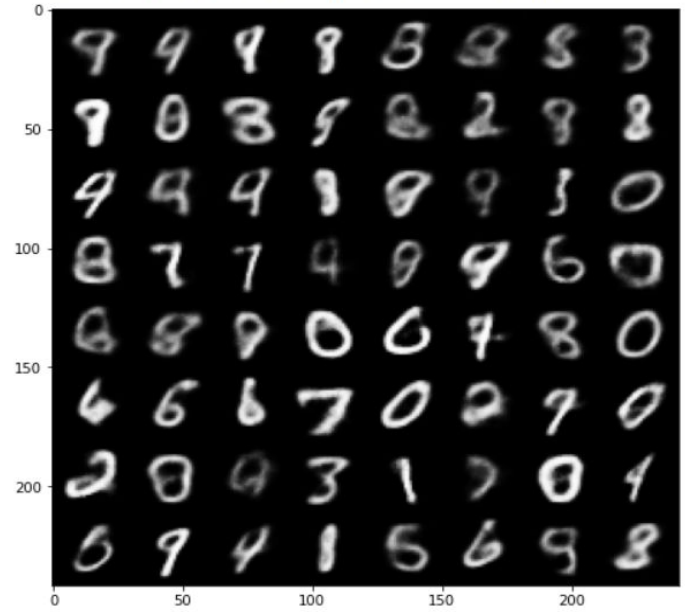
Generating Images as training Progresses :

Epochs for Training : 20

Before Training :

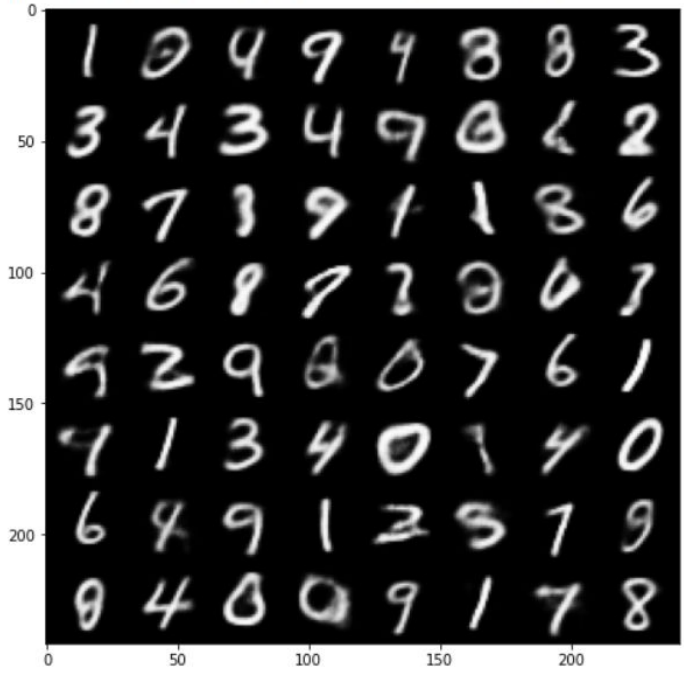


Epoch 0, Train Loss: 149.62, Test Loss: 117.71



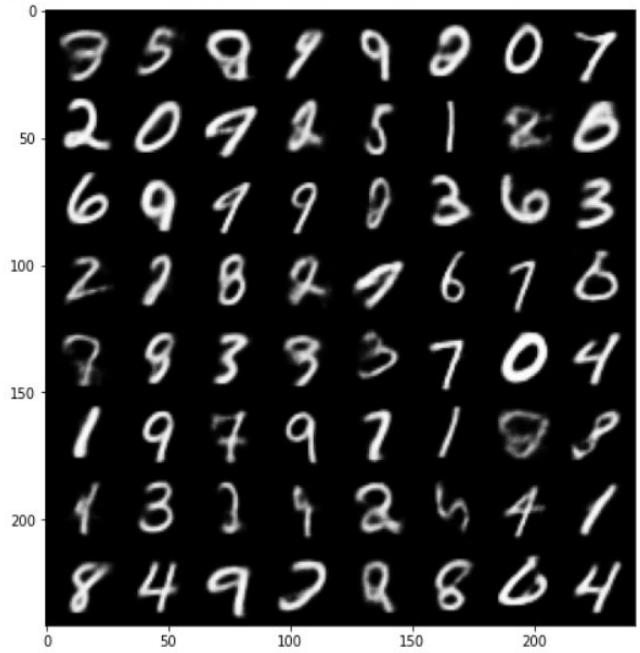
Epoch 5

Epoch 4, Train Loss: 105.21, Test Loss: 105.12



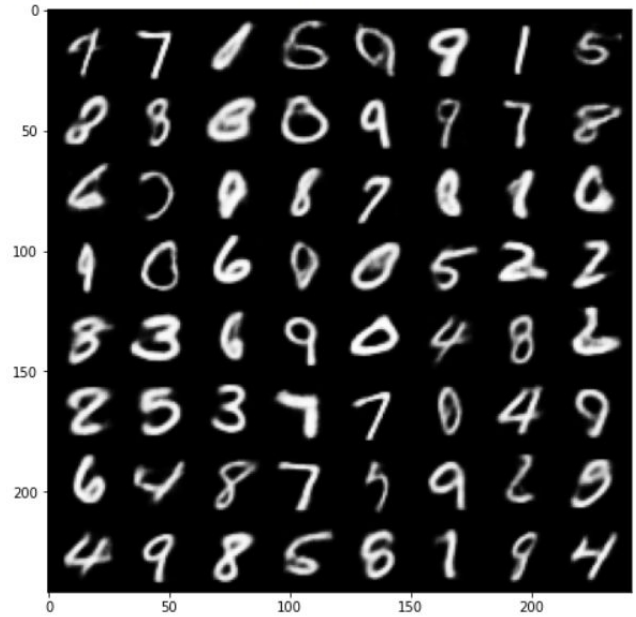
Epoch 10

Epoch 9, Train Loss: 101.13, Test Loss: 102.69



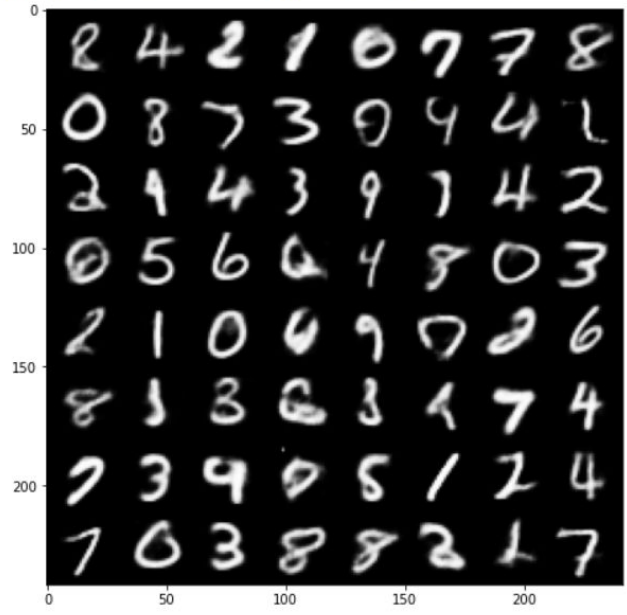
Epoch 15

Epoch 14, Train Loss: 98.98, Test Loss: 102.36

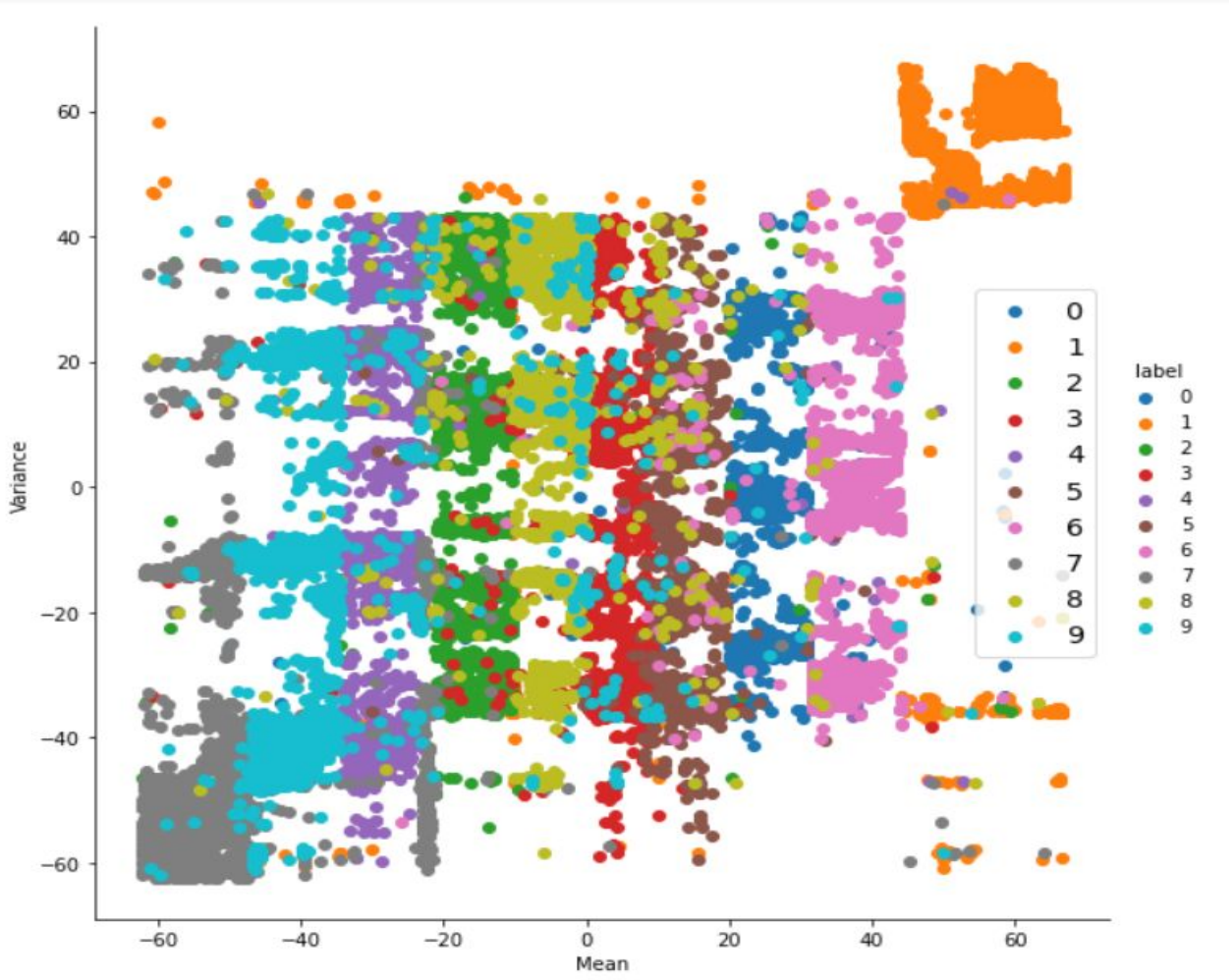


Epoch 20

Epoch 19, Train Loss: 97.41, Test Loss: 101.55



Visualization of T-SNE plot on Latent Dimension



As we can see in the Latent Dimensions, The data belonging to the same label are clustered together.

Latent dims = 10, concatenation of means and variance => for each sample 20 dimensions

```
print('Shape of Training data : ', train_latent_data.shape)
print('Shape of Testing Data : ', test_latent_data.shape)
```

```
Shape of Training data :  (60000, 20)
Shape of Testing Data :  (10000, 20)
```

Training SVM on above Latent data :

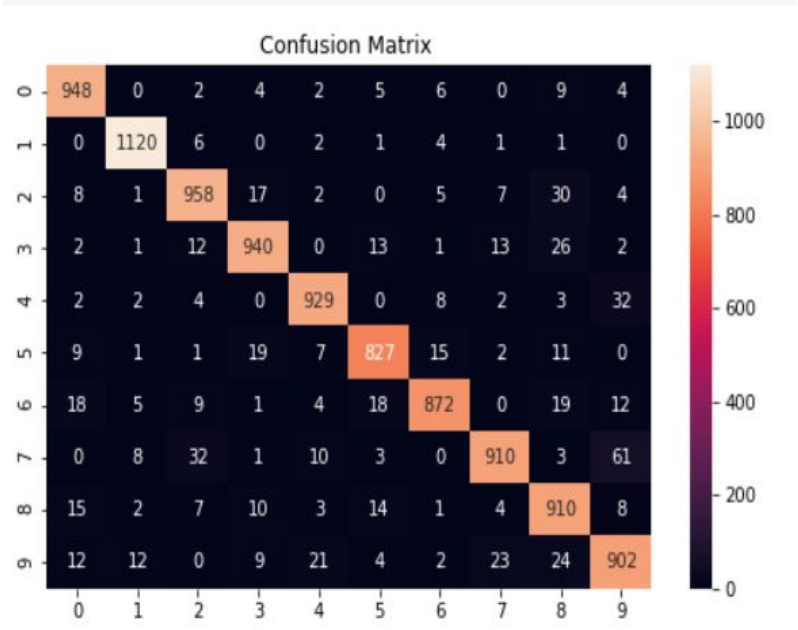
```
LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
          intercept_scaling=1, loss='squared_hinge', max_iter=1000,
          multi_class='ovr', penalty='l2', random_state=0, tol=1e-05,
          verbose=0)
```

Metrics :

```
print('Test Accuracy : ', test_acc)
print('Test prec : ', prec)
print('Test Recall : ', recall)
print('Test Data F - Score : ', f_score)
```

```
Test Accuracy :  0.9316
Test prec :  0.9316184285117484
Test Recall :  0.9309966925068466
Test Data F - Score :  0.9310271278772506
```

Confusion Matrix :



Conclusion :

With latent dimensions = 10 and using CNN based encoder and Decoder, with 20 epochs, we achieved good results.

The generated data is also good with very less noise. T-SNE plot shows that in latent dimensions, data belonging to the same class are clustered.

We achieved **accuracy of 93** on test data using SVM classifier. Precision and Recall are very good considering it is a 10 class classification model.