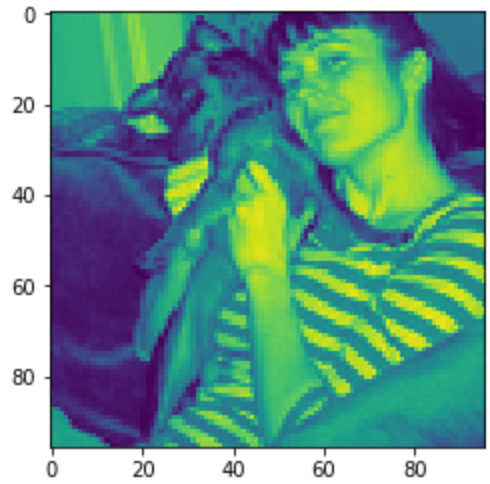
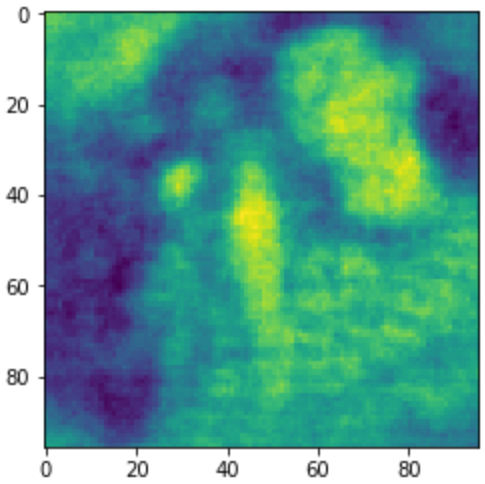
Question 1:
Full AutoEncoder:
Last 4 epoch loss learning rate 0.0001 and batch_size=100:
Loss: MSELoss:

epoch [96/100], loss:319.2539
epoch [97/100], loss:316.1426
epoch [98/100], loss:312.3979
epoch [99/100], loss:311.1474
epoch [100/100], loss:306.6875

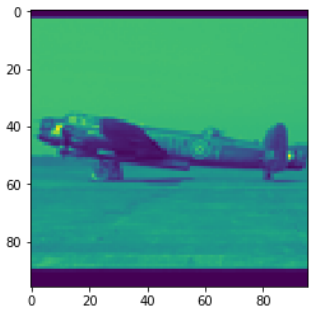
Real:



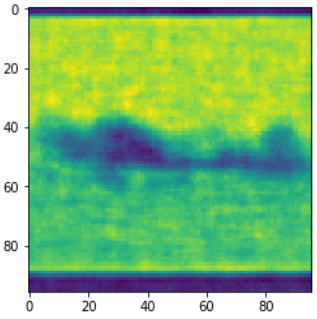
Stacked:



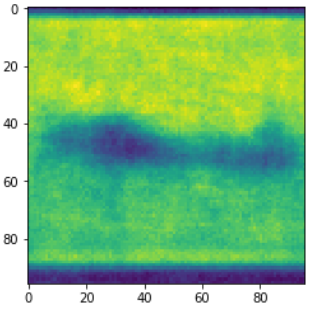
Real:



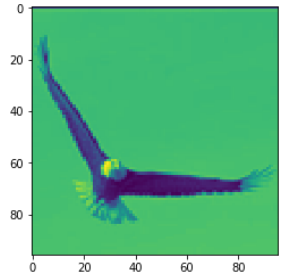
Full:



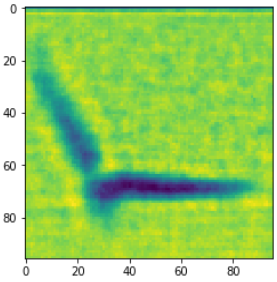
Stacked:



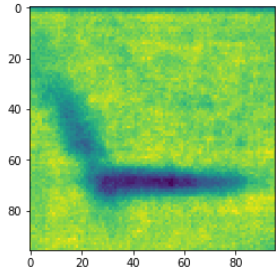
Real:



Full:



Stacked:

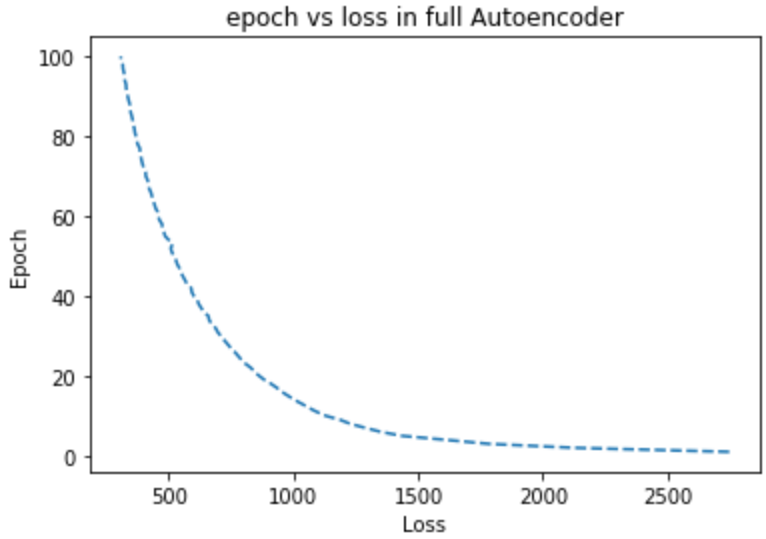


Stacked Autoencoder:
With same parameters as above:

It seems that Full AutoEncoder has learned representations better than stacked autoencoder. It may seem to happen that the weights which needs to update(from first layer) is not being updated in stacked autoencoder. As at any point of time only one layer is getting updated due to which it is not able to learn complex function. Due to it , the first is relatively better at reconstruction.

2) The accuracy on Full autoencoder when NN implemented in pytorch:

Without any norm (NN): 0.3758

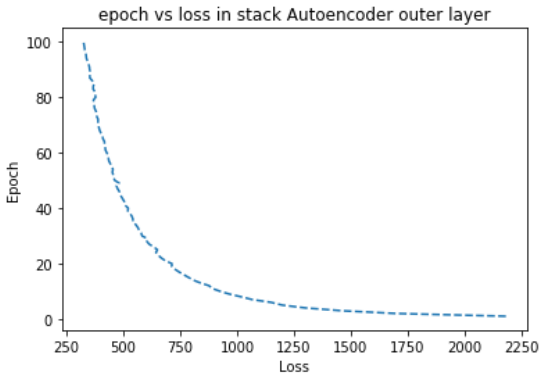


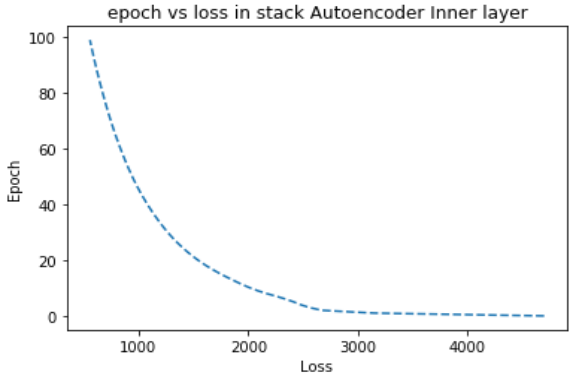
The accuracy on Stacked autoencoder when NN implemented in pytorch:

Without any norm (NN):

Outer Accuracy for outer layer:0.35623

Accuracy for inner layer 0.33417





Relatively better accuracy is with Full Autoencoder. It can be due to the fact that when the internal representation weights are changing, it's not having any effect in outer layer of stacked autoencoder. Due to this reason it won't be able to learn from loss of 2nd layer representation. It is justified in the accuracies provided above.

3)

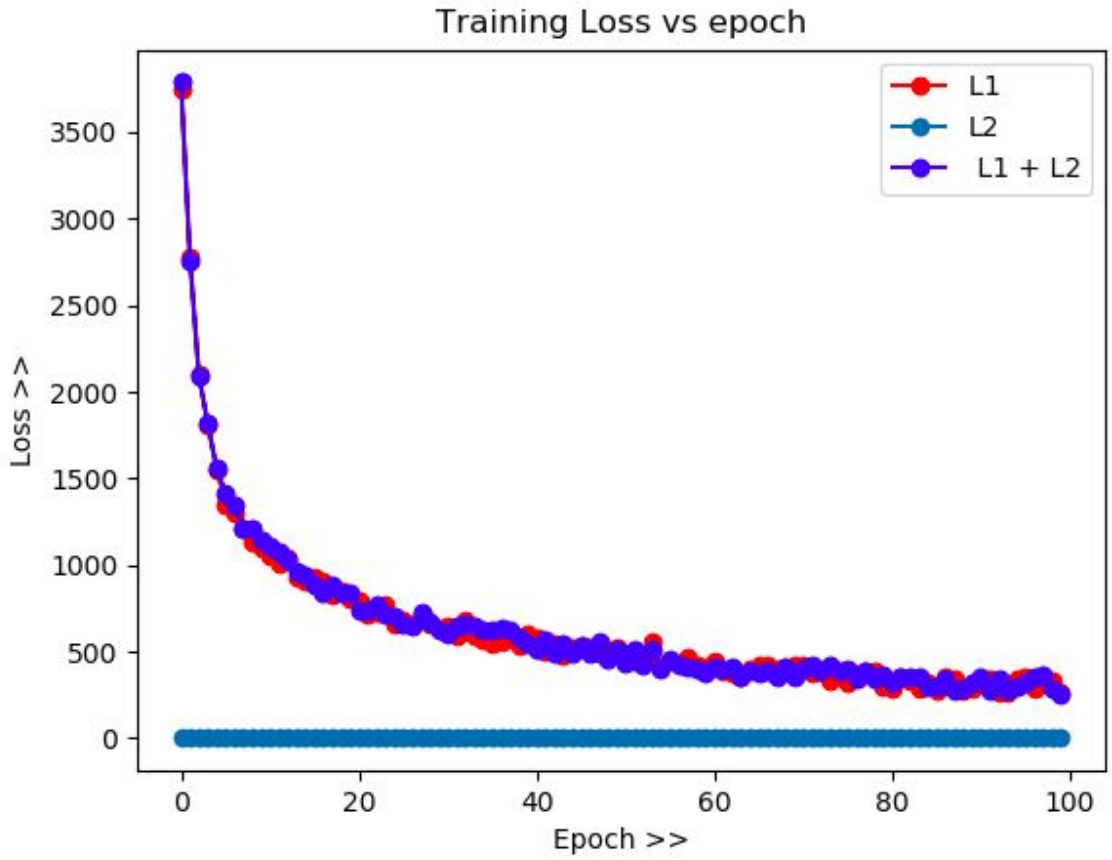
Full AutoEncoder		Stacked Autoencoder:	
With only l1-norm:	0.2645	With only l1-norm:	0.2855
With only l2-norm:	0.3734	With only l2-norm:	0.3457
With both norms:	0.2354	With both norms:	0.3244

The reasons for these accuracies are that with l1-norm, there might be possibility that noisy features are having more importance and their weight is not becoming 0. So because of which we got less accuracy, but with the case of l2-norm, as it distributes the weights , due to which the importance in decision making of features is not entitled to one and therefor accuracy is getting increased.

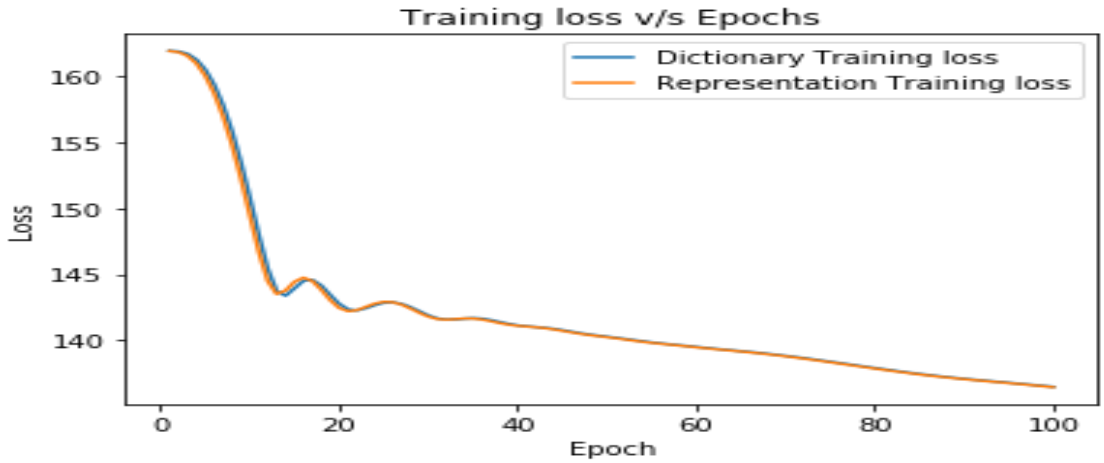
The accuracy when L21 is applied should come in between of both L1 and L2 combined but here it is in the case of stacked Autoencoder.

4) Supervised Autoencoder Accuracy: : 0.3875
With L2-norm : 0.3836

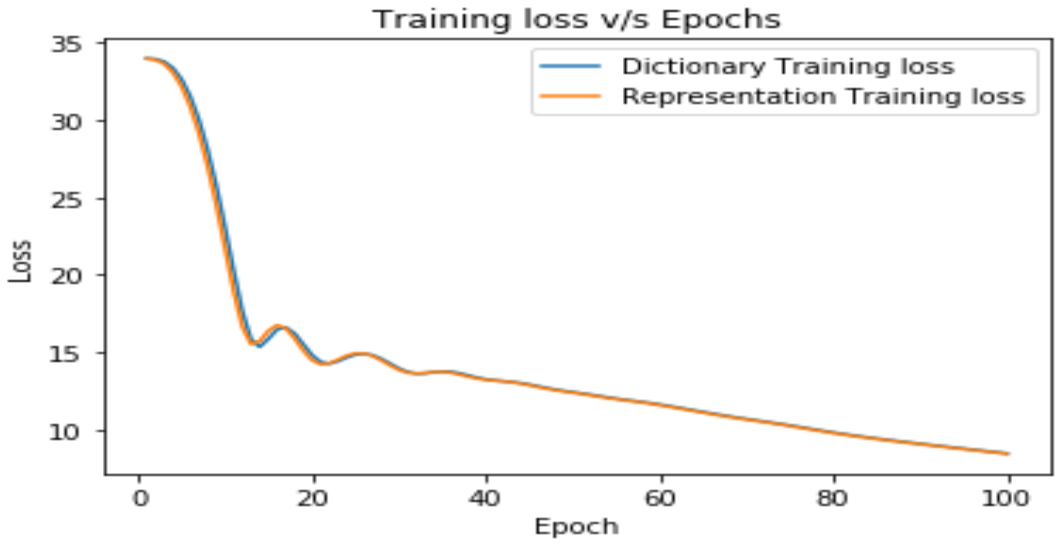
Here Loss is max in all the above cases as loss due to supervision is also added in the network. Due to which it might be penalizing more to the learned weights.Therefore accuracy is high in this case than earlier models.



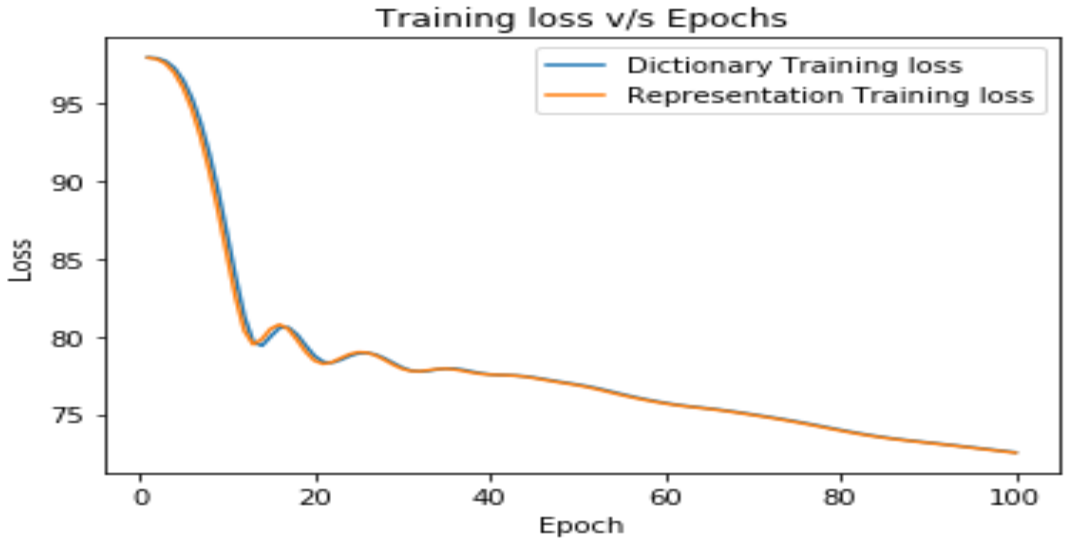
Question 1)
With sparsity factor =1, epochs =100, atoms =1000



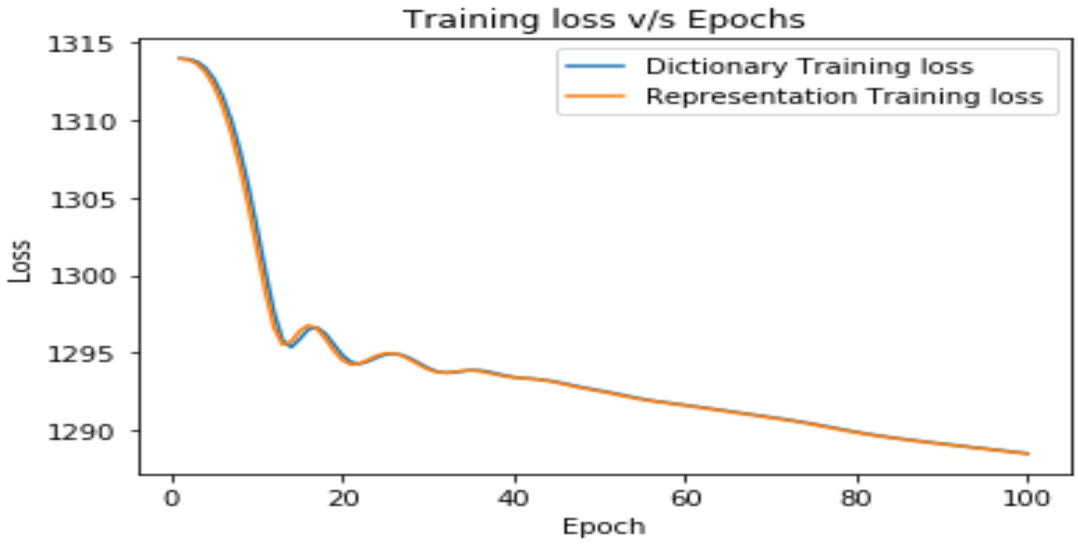
With sparcity factor=0



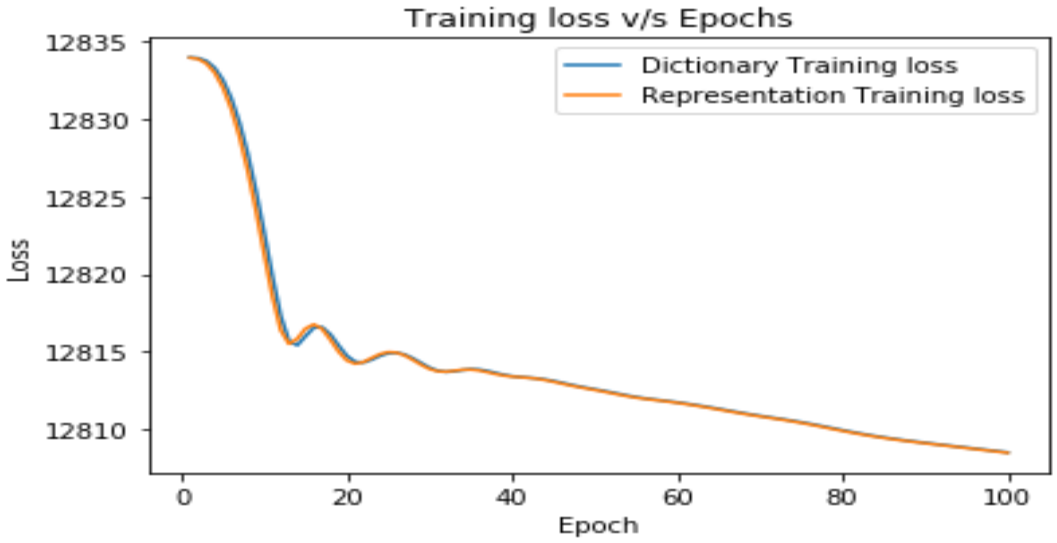
With sparcity factor = 0.5:



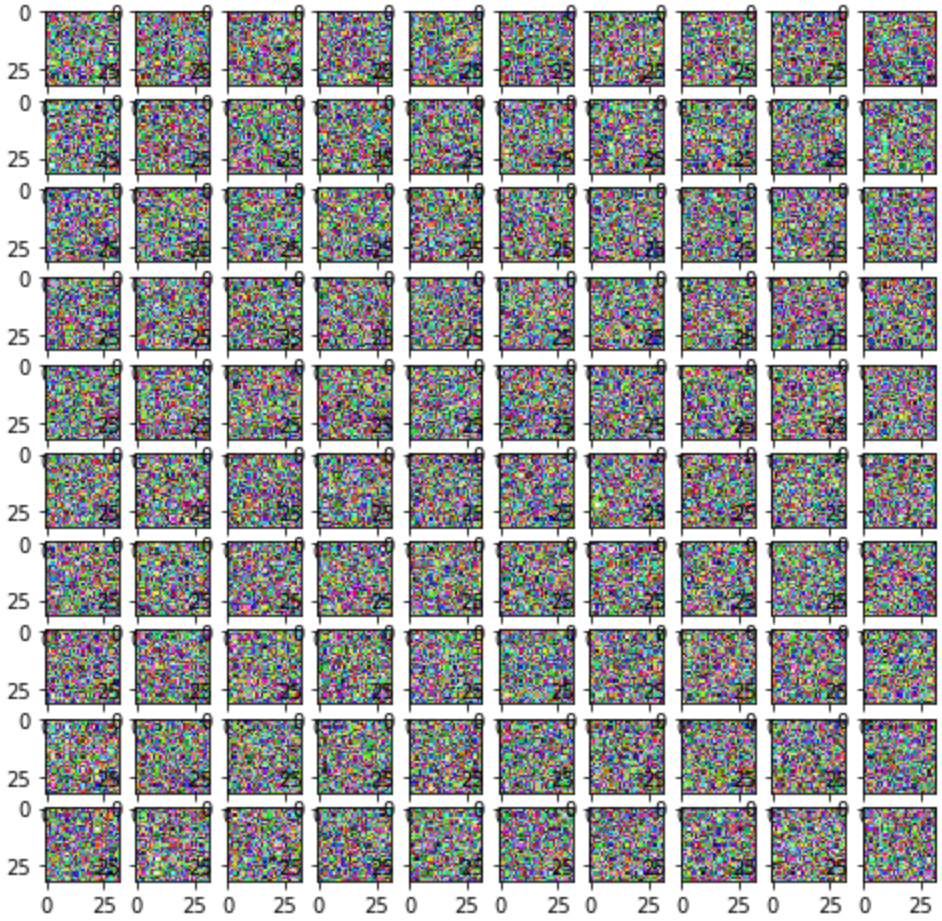
With sparcity factor: 10



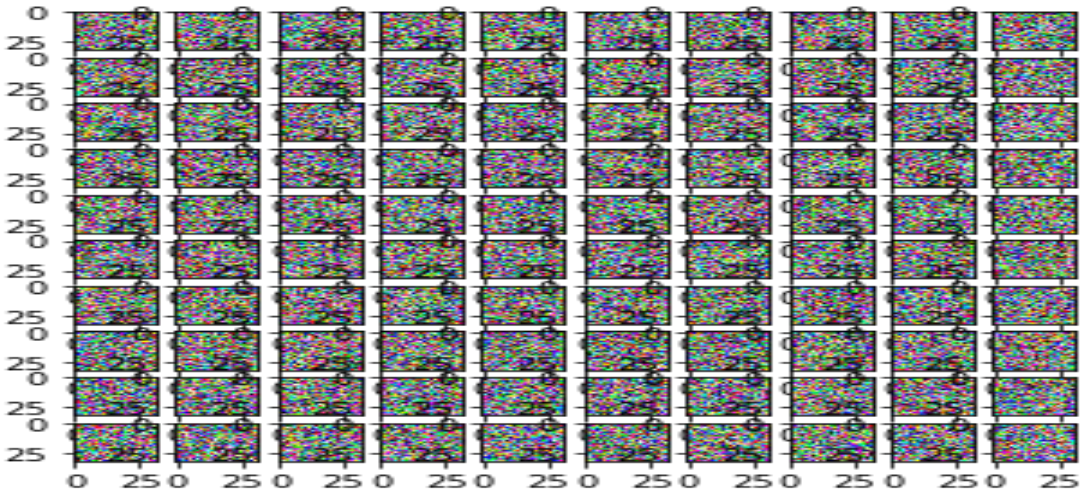
With Sparcity factor 100:



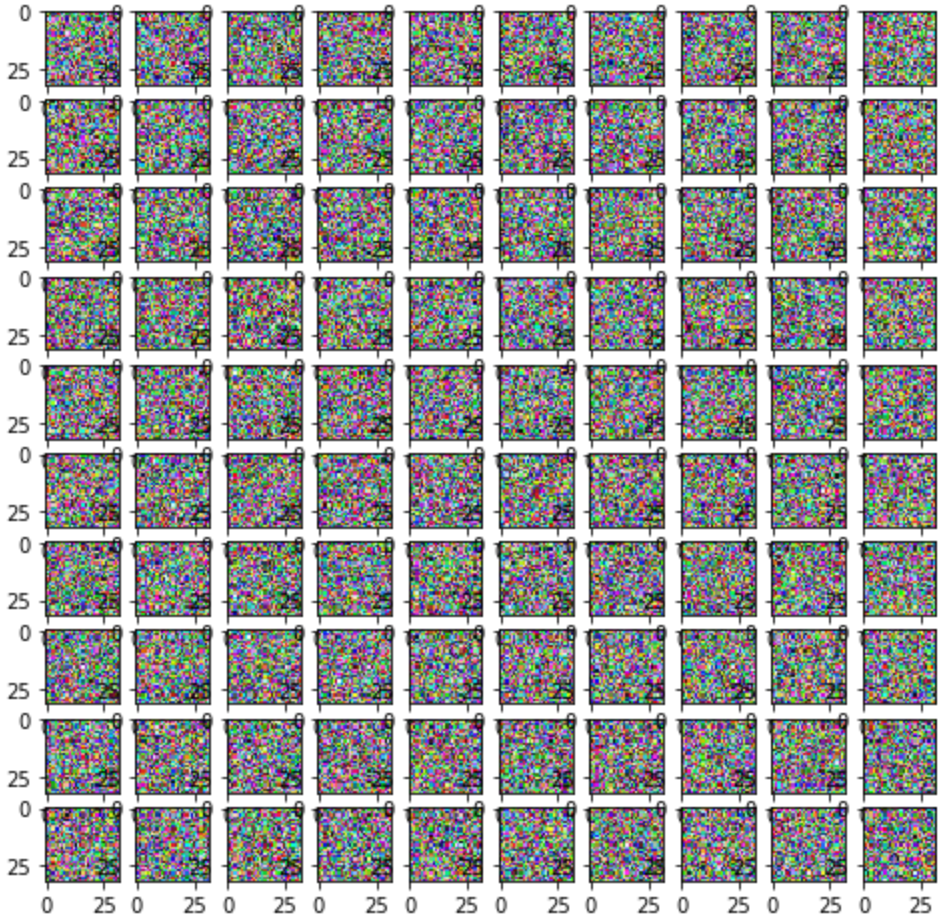
Constructed Images at lambda 1 :



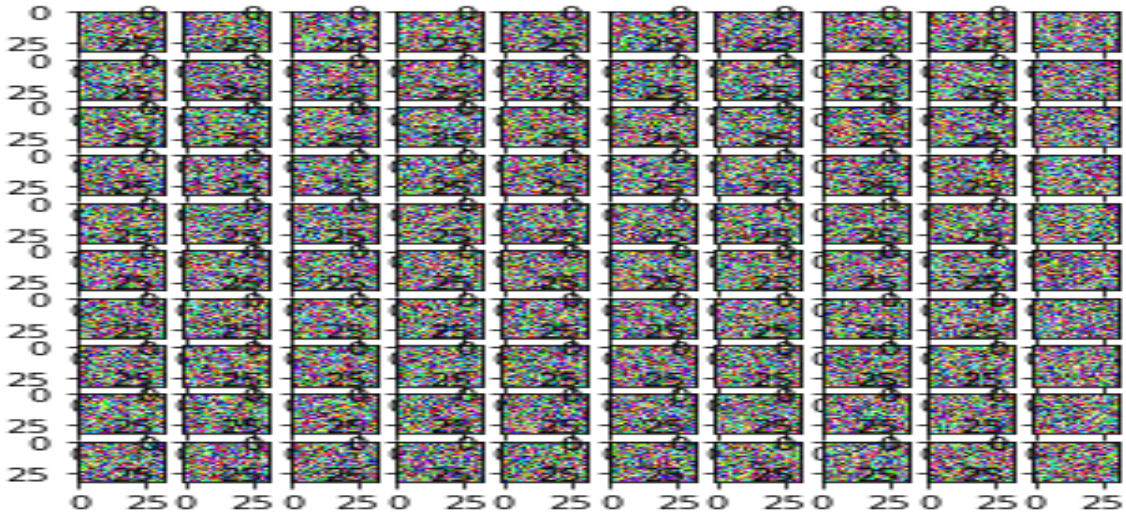
Atoms:



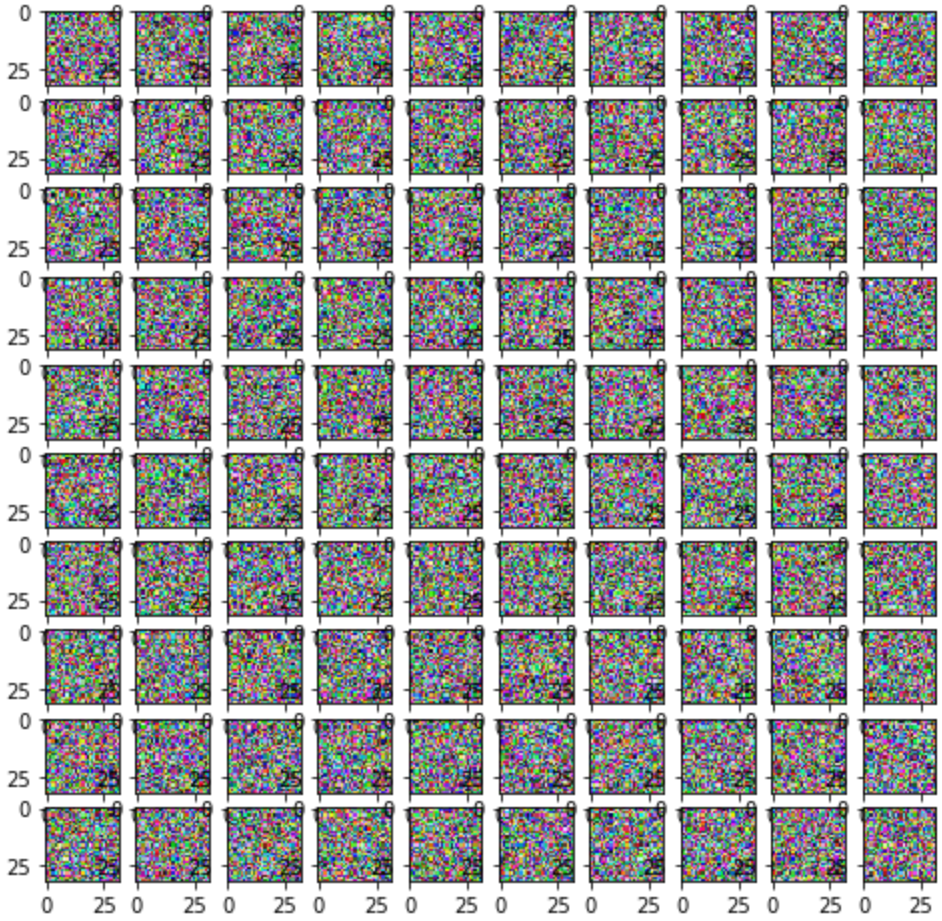
Reconstructed with 0 lambda



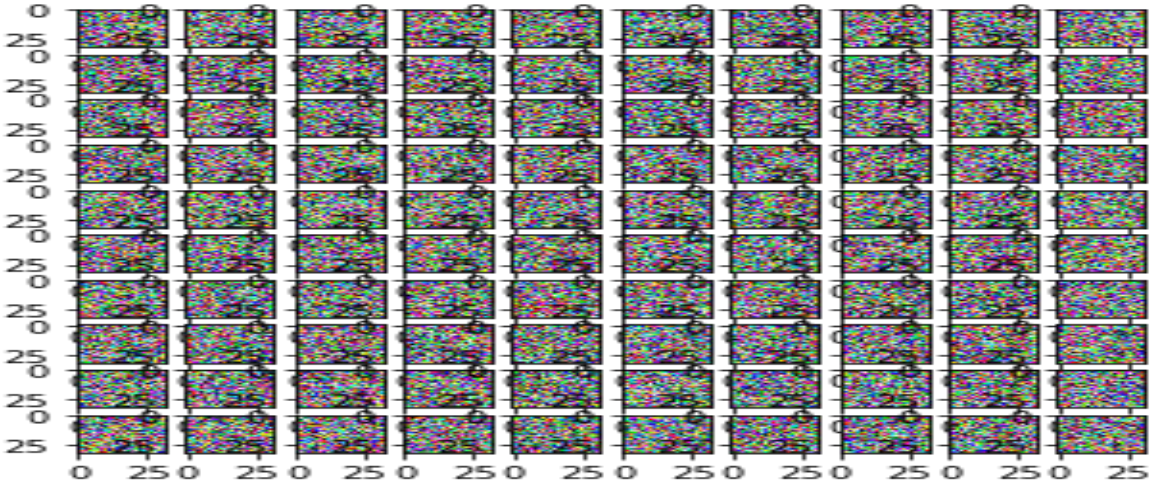
Atoms:



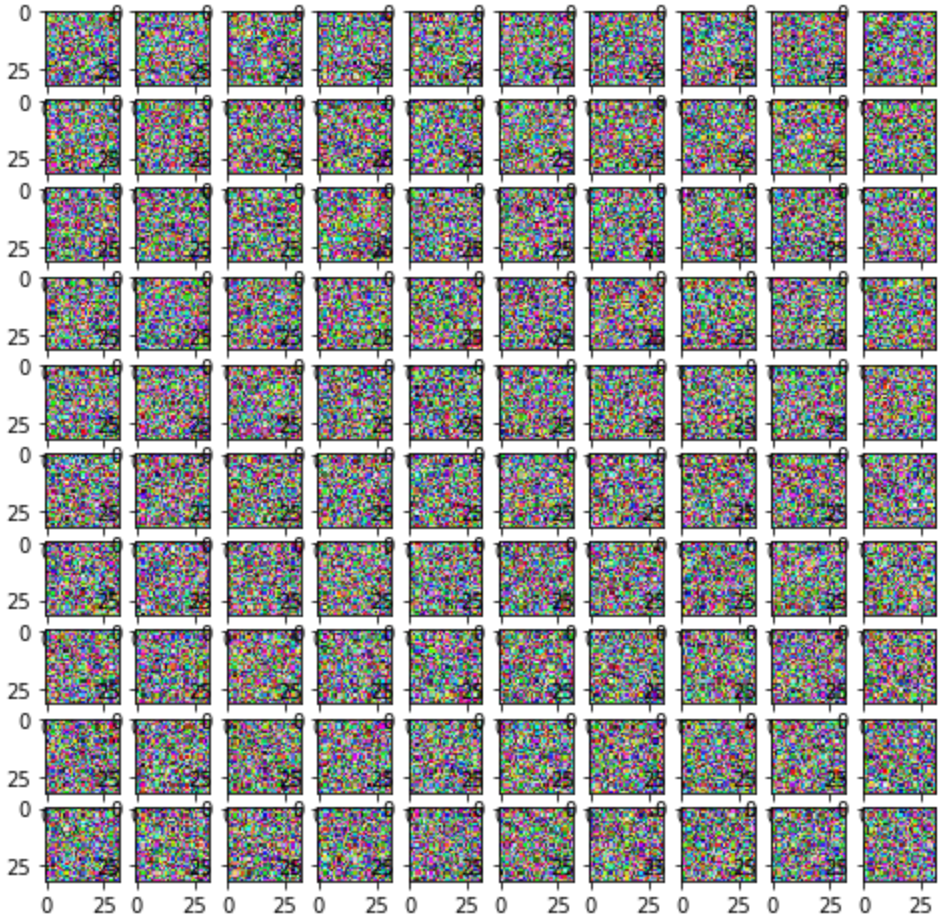
Reconsntructed with 0.5 lambda



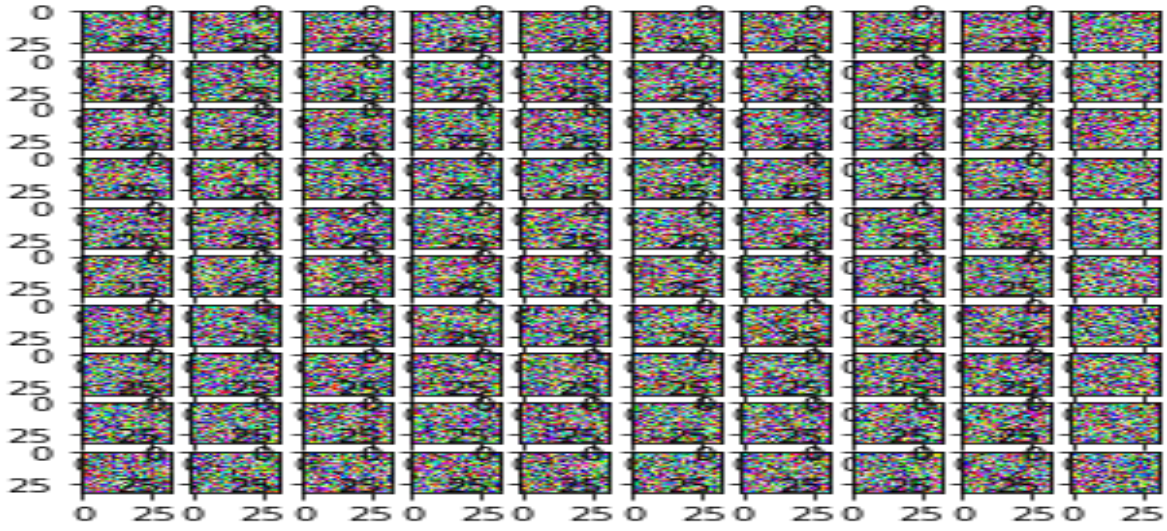
Atoms:



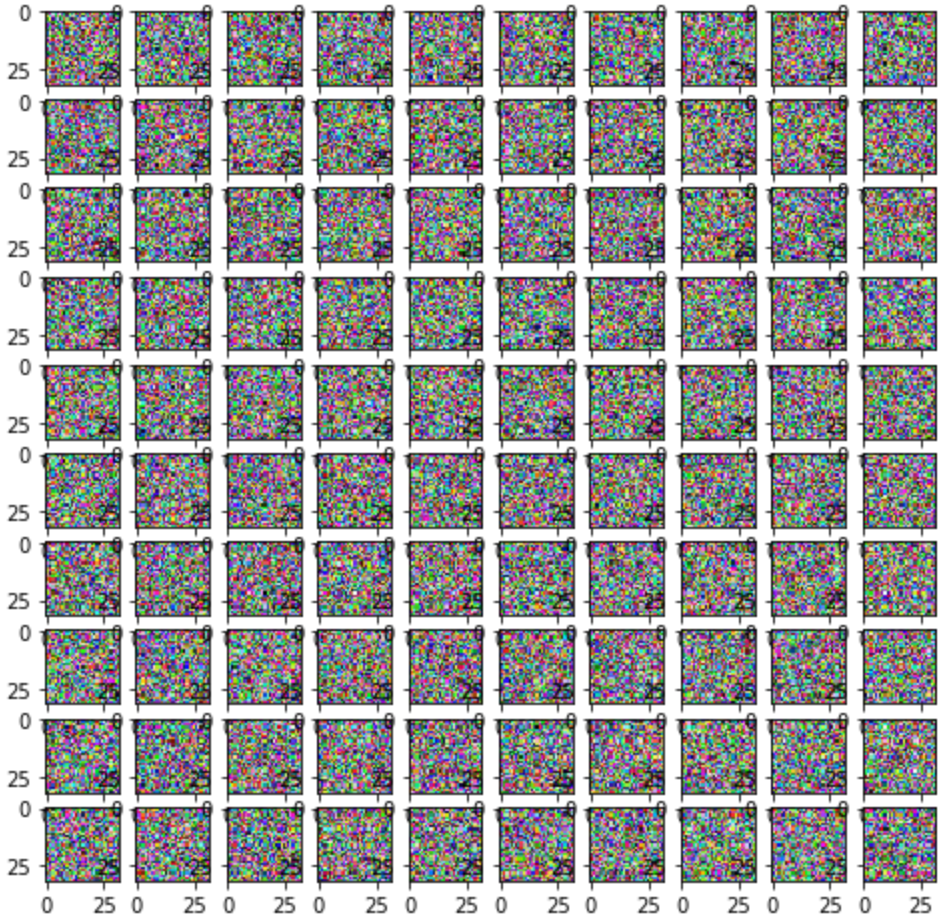
Reconsntructed with 10 lambda



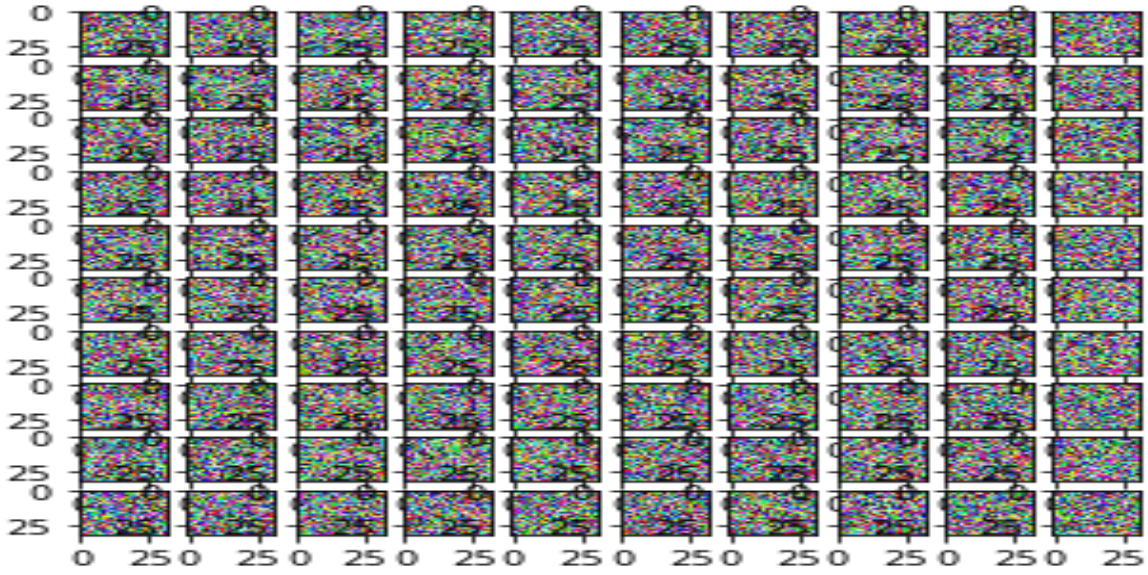
Atoms:



Reconstructed Images with 100 lambda



Atoms:



Nothing can be said about any dictionary be looking at the pictures but by seeing loss, we can say that with lambda 0, the loss was least. Due to which it has been chosen for Classificaion purpose.

4)Accuracy was 10.26%. Which was calculated on model with sparsity 0. The reason could be that due to that Dictionary is unable to learn representation.