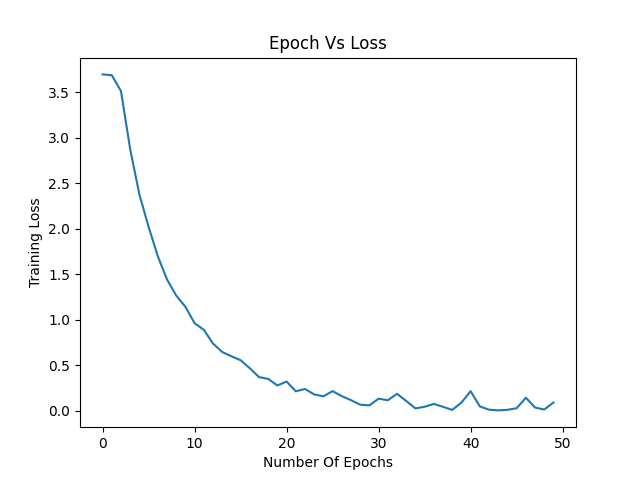
# Report on Model Inversion Project

### PET 2021

In model inversion attack the adversary uses a convolutional neural network model tries to invert the input data by using the labels. In our attack we use some face image data from AT&T faces dataset (<https://git-disl.github.io/GTDLBench/datasets/att_face_dataset/>) . The dataset contains 400 sample images where we can see 40 subjects each having 10 different images. All the images are in PGM format and in greyscale.

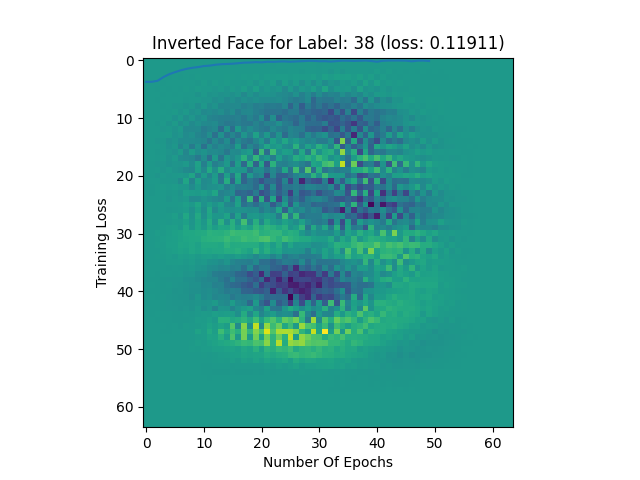
For our target model we split the dataset randomly in 90:10 ratio of training and testing dataset. Our target model consists of 3 layer convolutional neural network. The model uss kernel size of 2 and filter size of 32,64,128. For training the target model we run 50 epochs with learning rate of 0.001. We used Adam optimizer and calculated the cross-entropy loss. We can see our training loss in the following image-



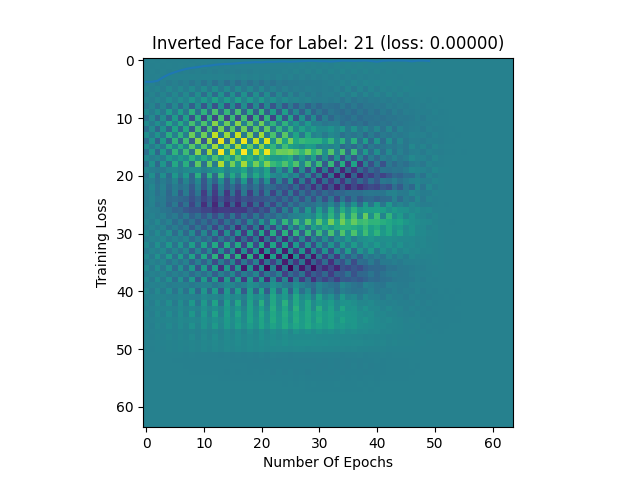
After training we used our testing data to evaluate the model. Using the AT&T faces dataset on our target model we achieved test accuracy of 90.00 %.

After that we built our attack model where we took our target model and set it to evaluation mode so no parameters are further updated. We then sent an empty image and cross check the probability with each image label. We use the calculate probability and create a cost function using 1 – probability. Then we back propagate and use the MI\_algorithm to generate the input image having those specific features. We run 1000 epochs for our attack. Finally, it will try to invert some of the faces from their label from the trained model using the MI-Face algorithm. It then creates a folder called inverted\_faces, which will have the generated inverted faces and their loss. We can see some of the generated images for the specific labels and their loss,

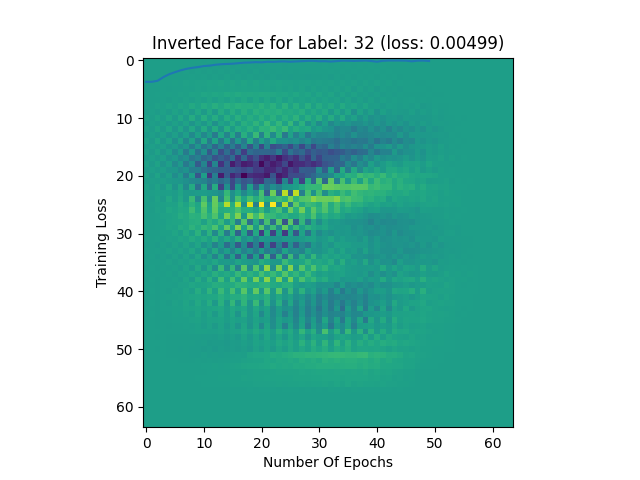
For label 38 we can see the following generated vs the input image,



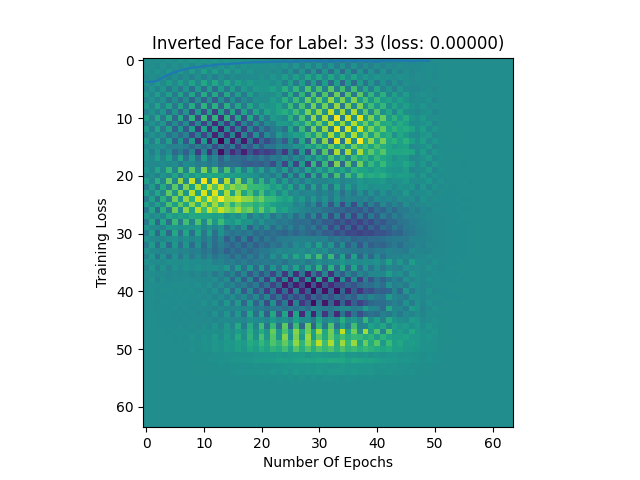
For image 21



For image 32



For image 33



From the above examples we can see that we somewhat generated an input image containing some related features to the original input images. As we implemented very basic of the proposed algorithm for model inversion the generated images are not accurate. We can achieve more accurate images using more complete version of the algorithm.