Federated Learning-Based 3D Medical Image Compression

In propose work we are employing Optimal Multi-linear Singular Value Decomposition (OMLSVD) and deep auto-encoders to compress medical 3D images. 3D medical images contains accurate information about diseases but this require heavy storage and to reduce storage size many compression algorithms were introduced such as PCA, DWT, JPEGCompreesion2000 and many more. All existing algorithms decompress image quality is very low.

To enhance 3D compression image quality we are combining OMLSVD and auto-encoder where OMLSVD will be utilize to compress image and then compress image along with normal image will get trained with auto-encoder algorithm to decompress compress image with high quality.

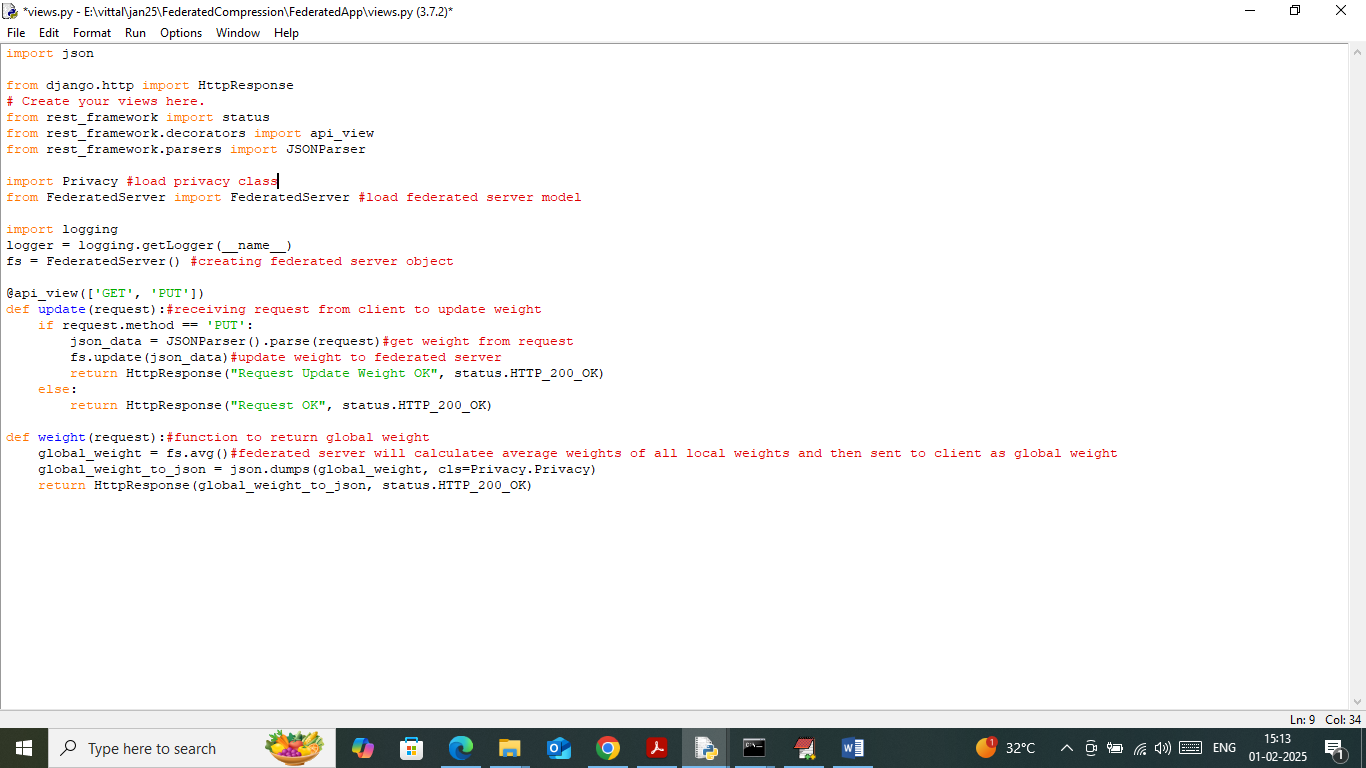
OMLSVD and auto-encoder required less storage image compare to existing algorithms. Traditional training of Auto-Encoder model require sharing of datasets or weights with all medical clients which may leak data of one medical client with other medical client.

To overcome from above data privacy and leakage we are employing Federated Learning technique where all clients will trained model on their datasets and then send their model local weights to Global federated server. Anytime any medical client can request federated server to get global weights of all clients and then perform prediction. So by employing Federated server we can get global dataset weight without publishing and leaking dataset so privacy will be achieved.

To train and test existing and propose auto-encoder algorithms we have used 3D chest dataset which can be download from below link

<https://www.kaggle.com/datasets/constantinseibold/anatomy-in-chest-x-rays-pax-ray>

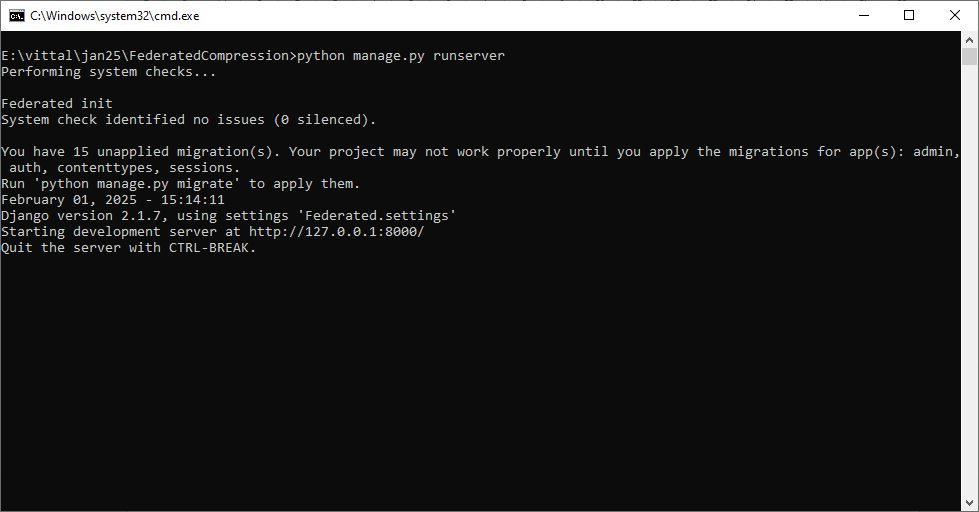
We have designed following Federated Server to manage local and global weights



In above screen read red colour comments to know about local and global model weights of Federated server with privacy.

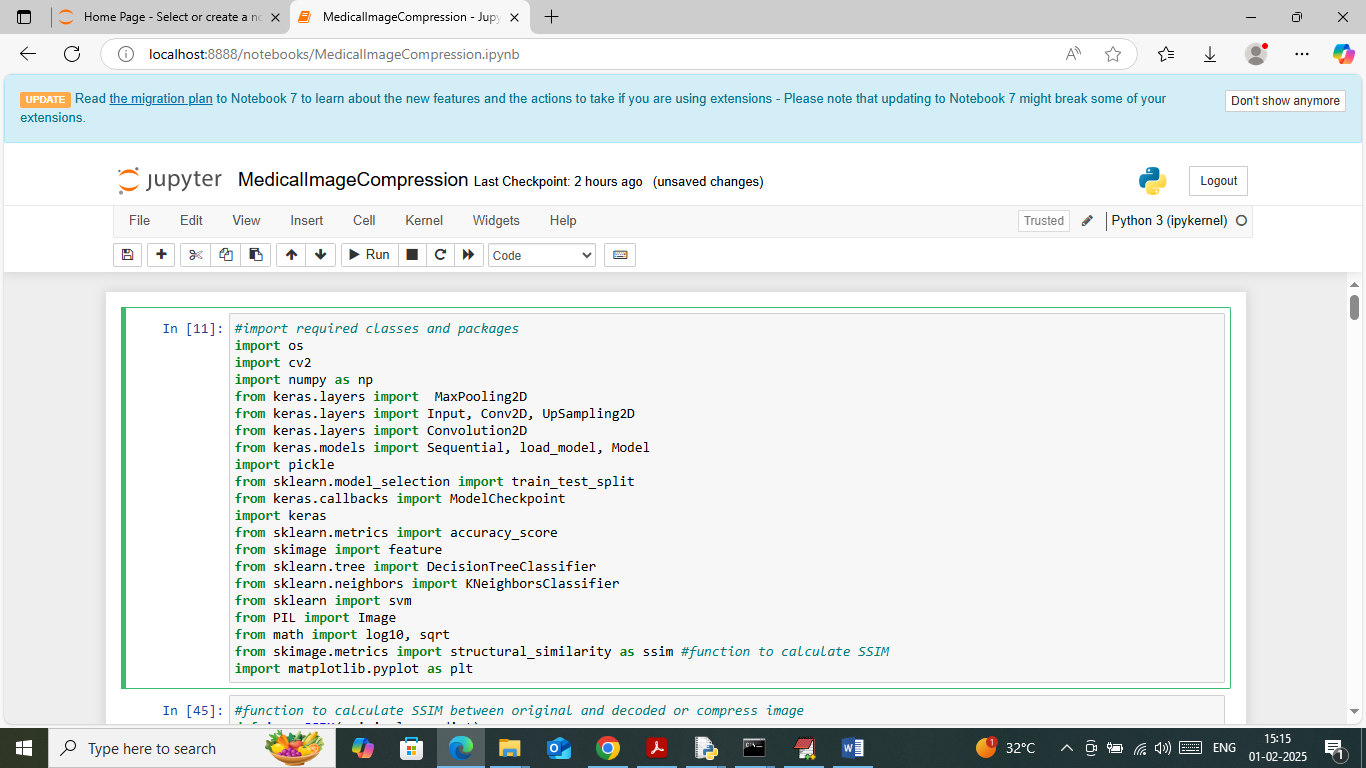
SCREEN SHOTS

First double click on ‘runFLServer.bat’ file to start federated server and then will get below page

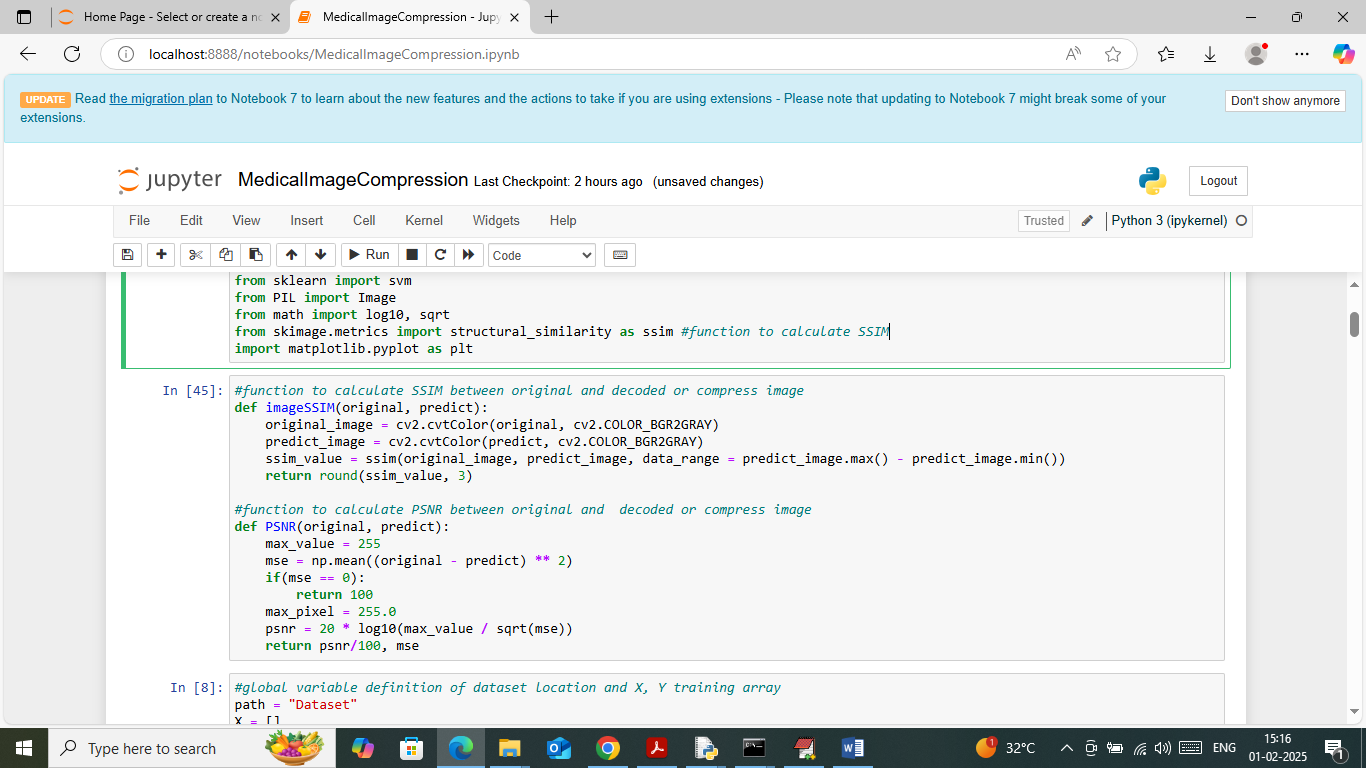


In above screen Federated server started and now double click on ‘runJupyter.bat’ file to start JUPYTER notebook and then can see below output screens.

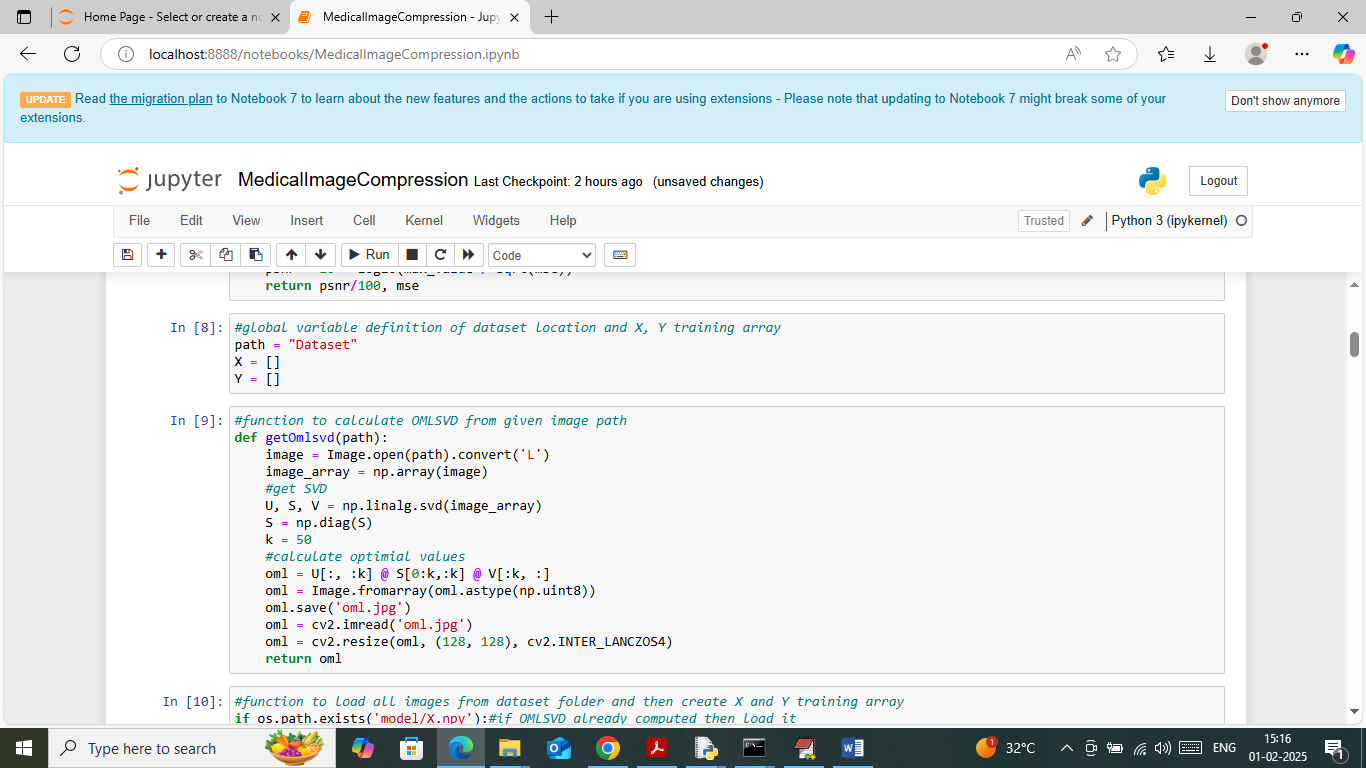
We have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments



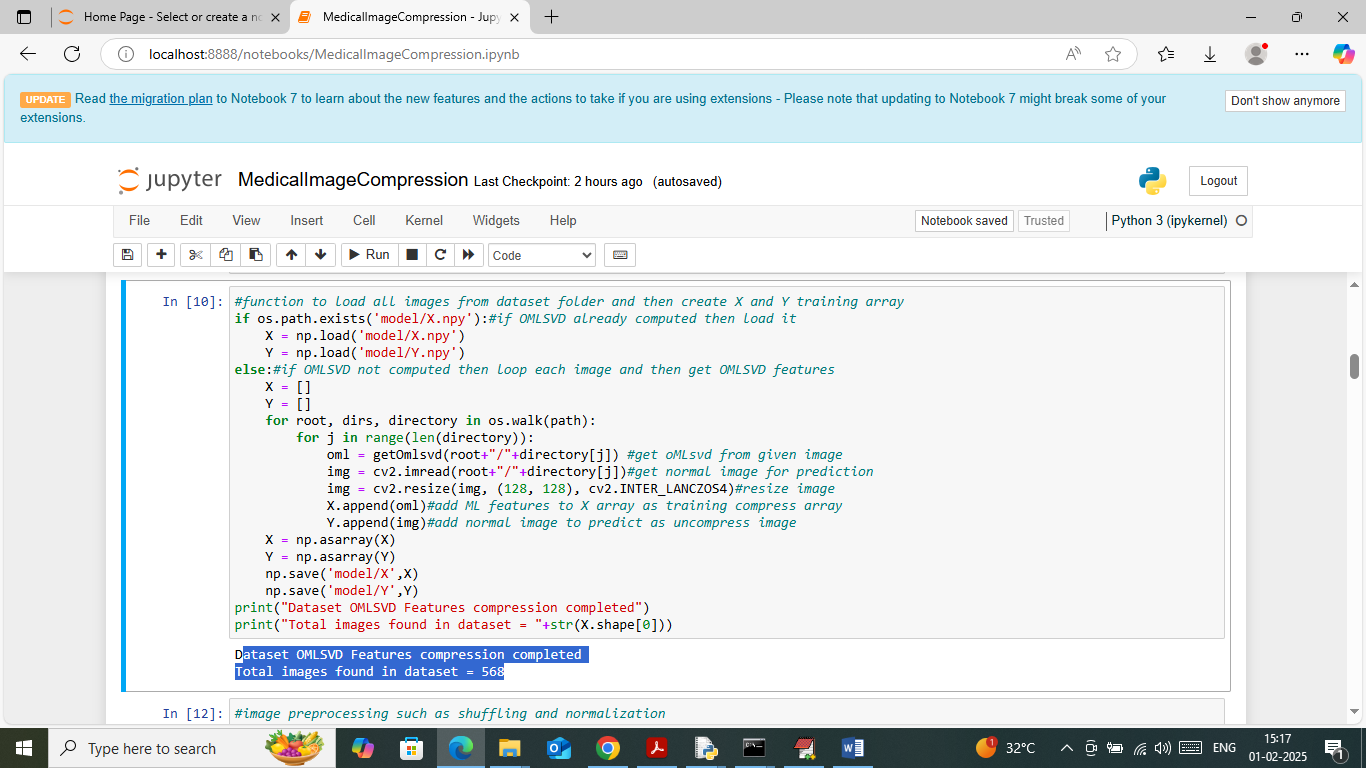
In above screen importing required python classes and packages



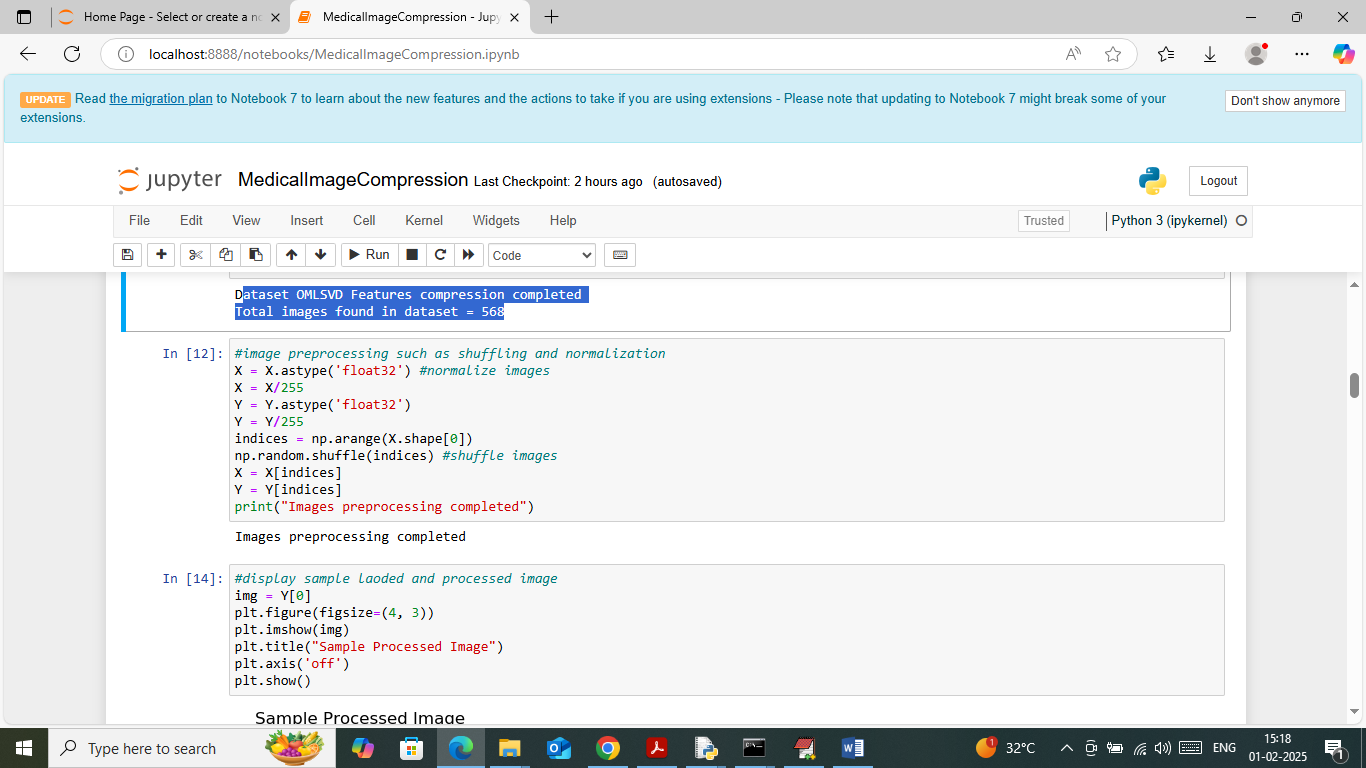
In above screen defining function to calculate SSIM and PSNR values



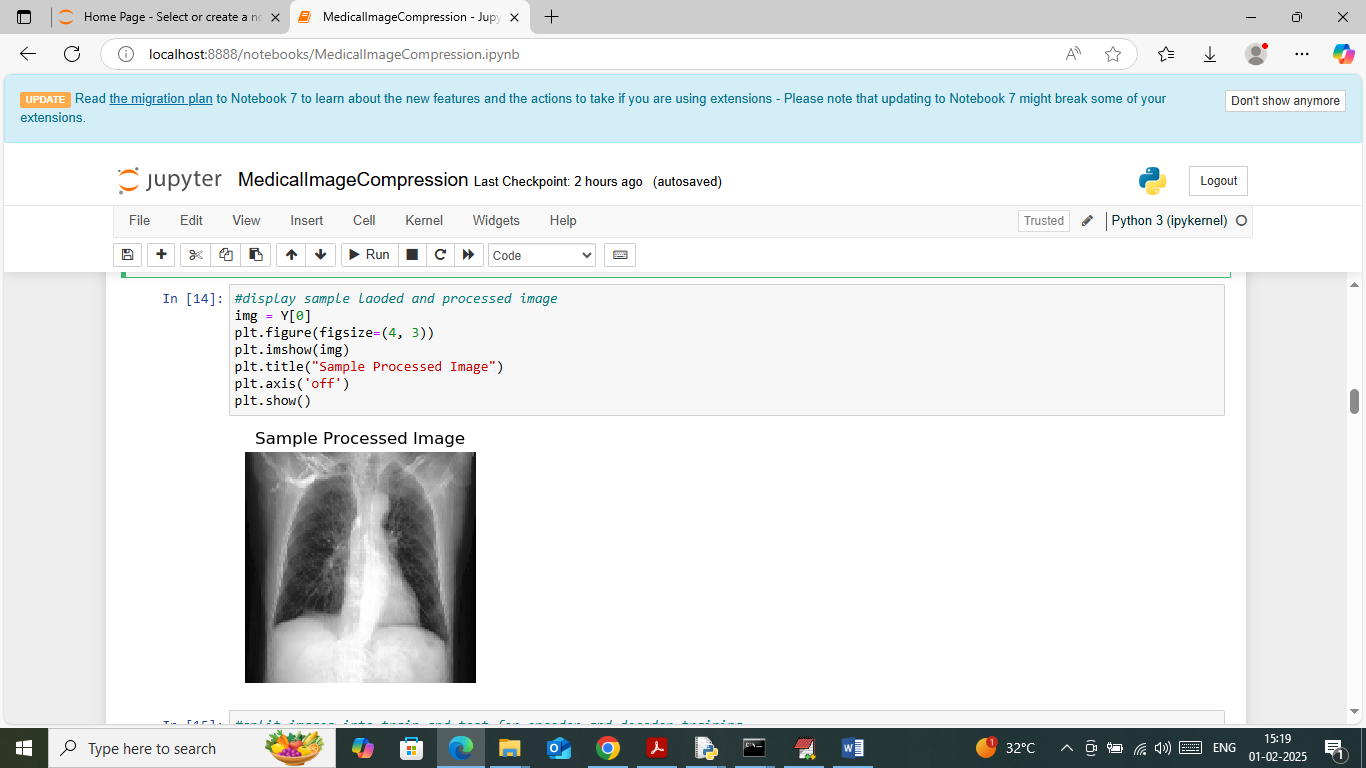
In above screen defining function to compress image using OMLSVD



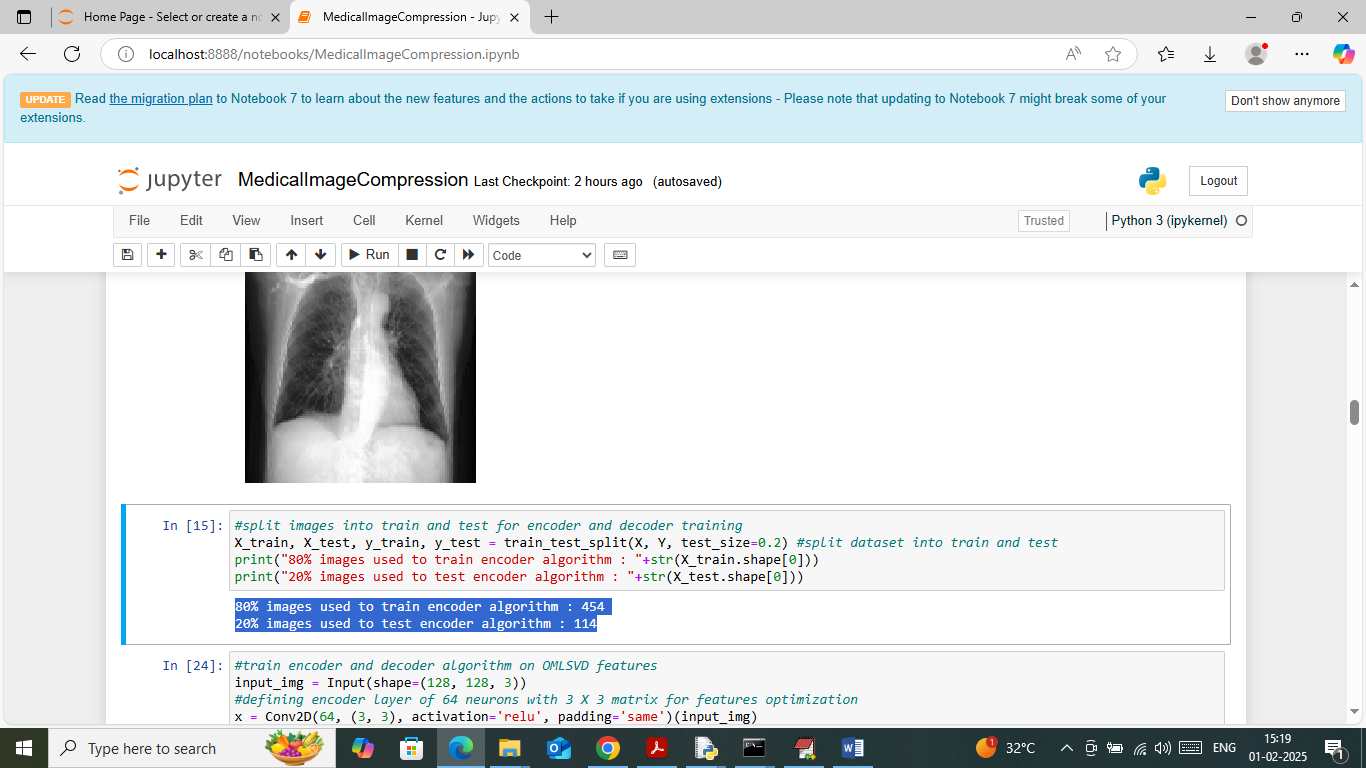
In above screen looping and reading all images from dataset and then compressing using OMLSVD and then creating training array. In above screen can total number of images loaded



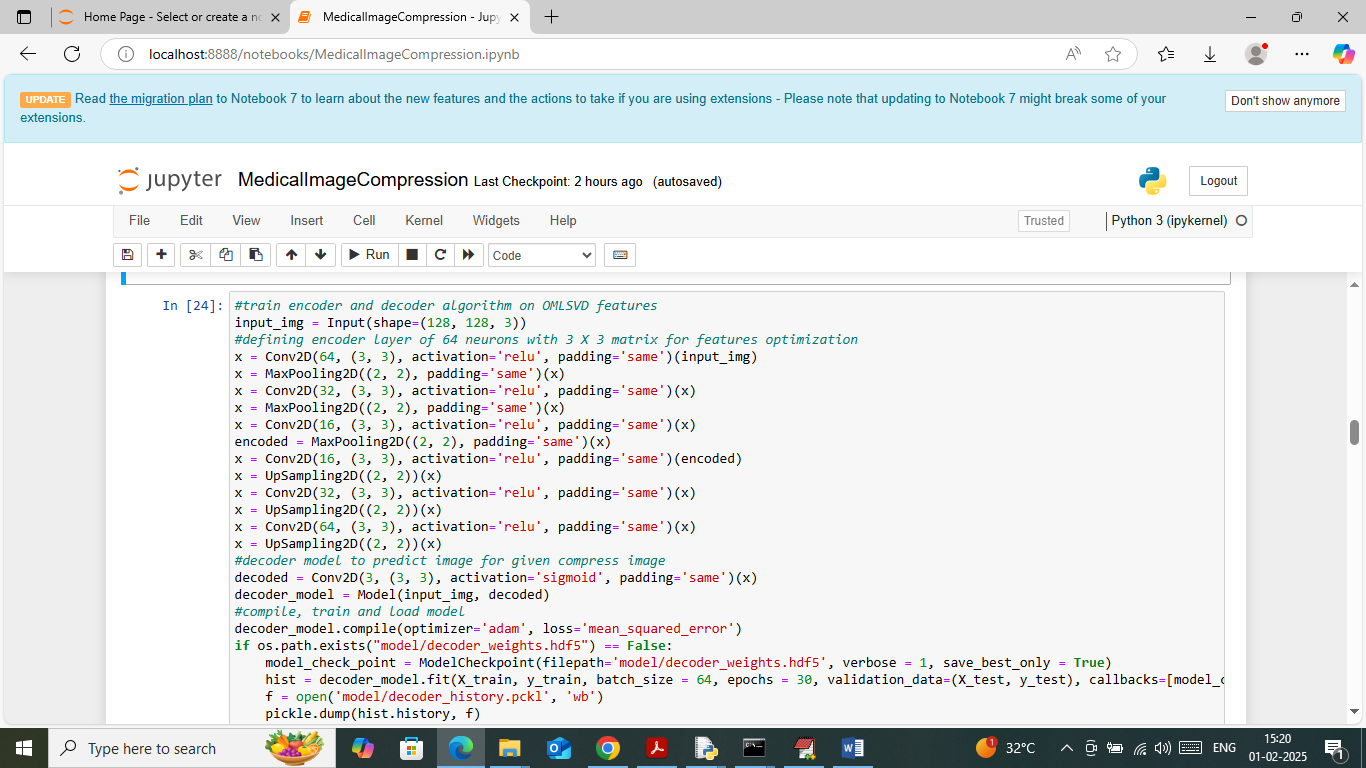
In above screen applying image processing technique to normalize and shuffle images



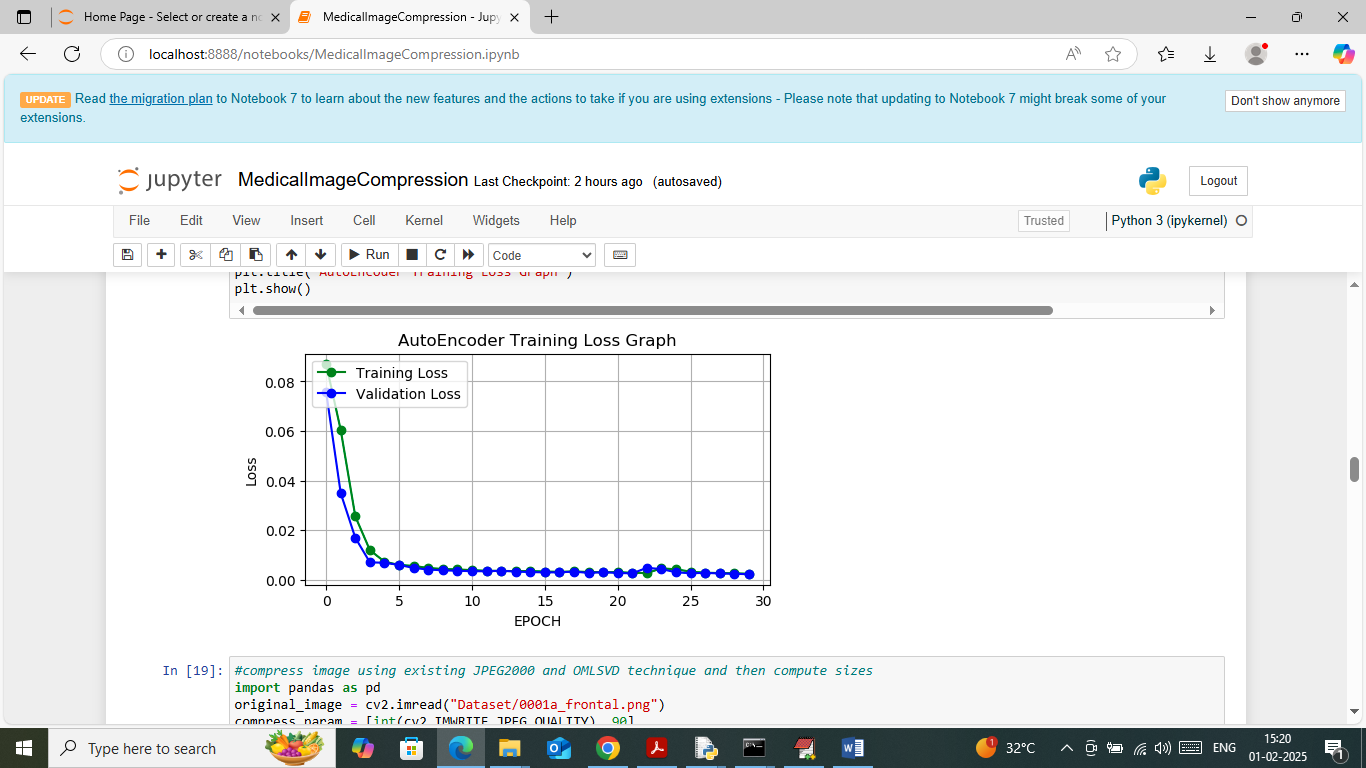
In above screen displaying sample processed image



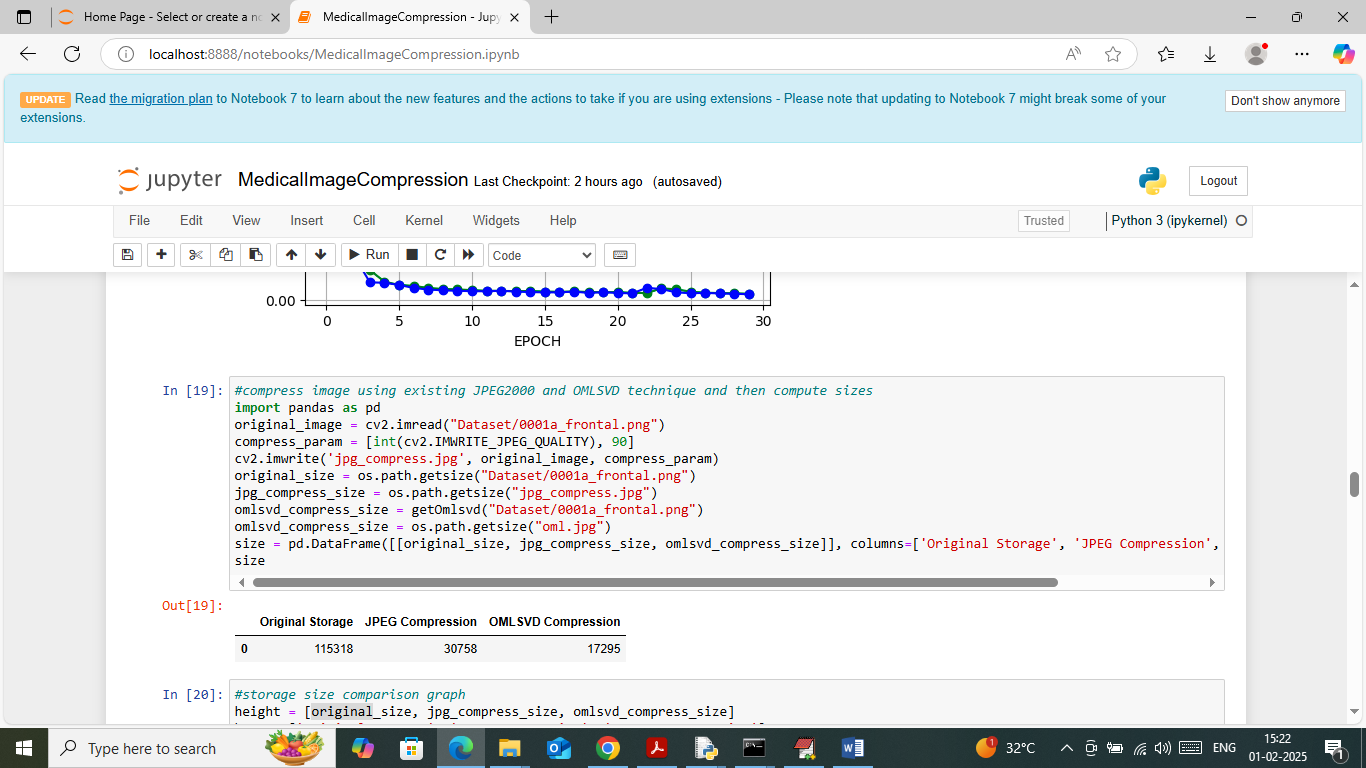
In above screen defining function to split dataset into train and test where application using 80% images for training and 20% for testing



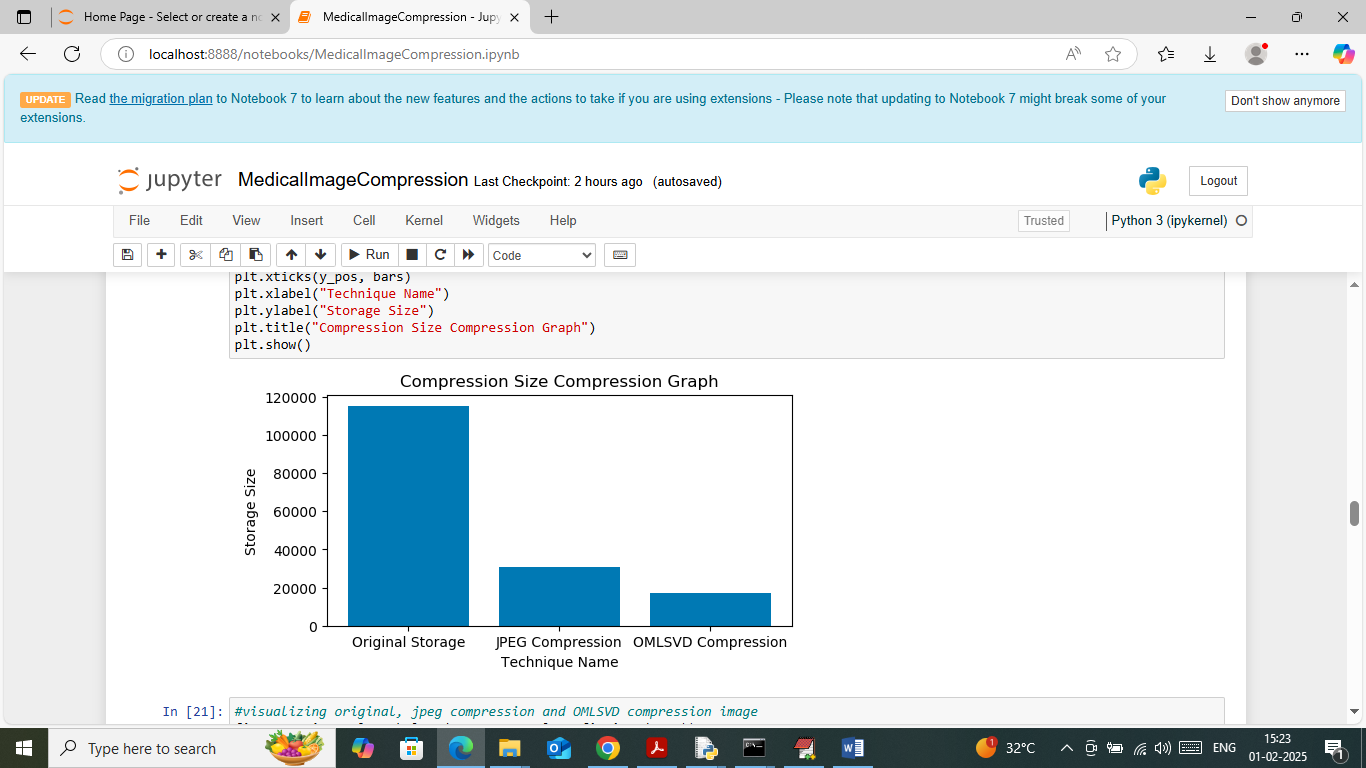
In above screen defining auto-encoder model to train compressed images and then will get below output



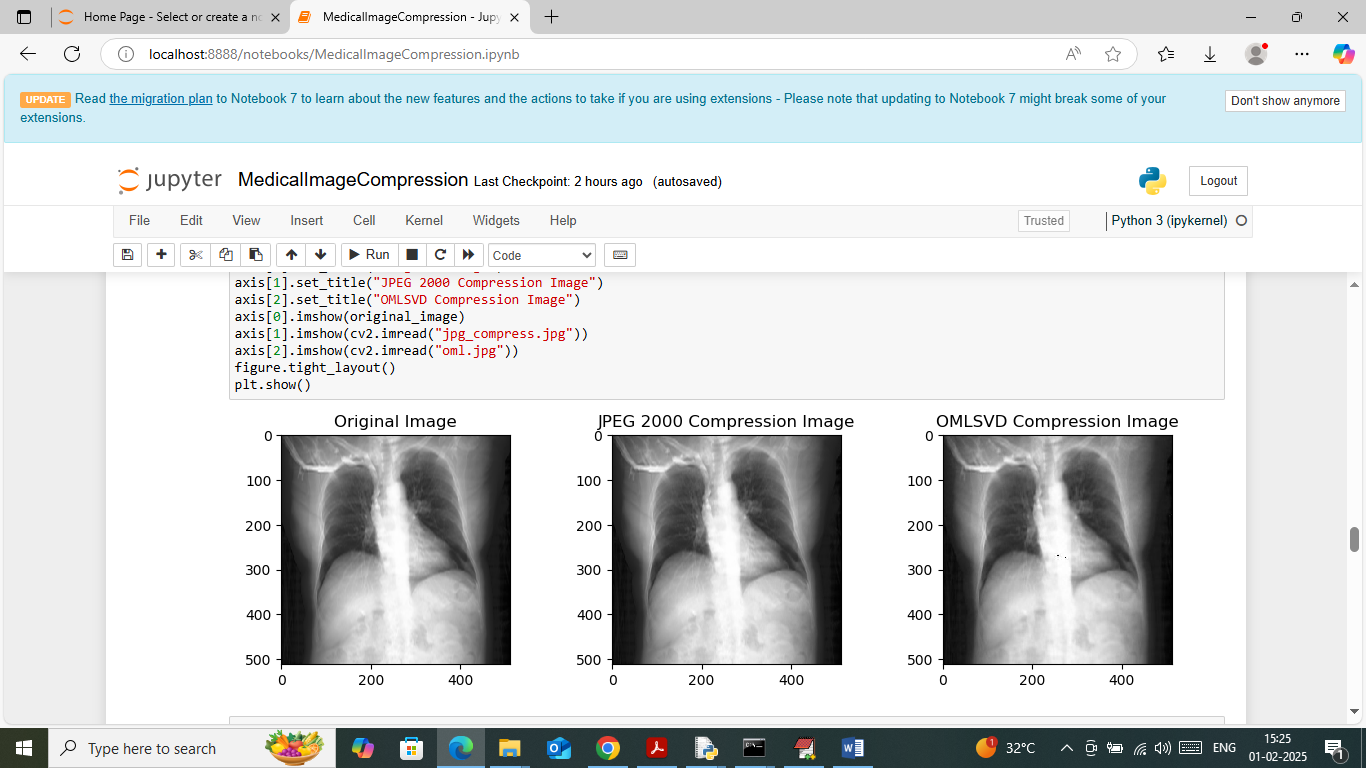
In above screen auto-encoder model training completed and in graph can see auto-encoder model training and validation loss. In above graph x-axis represents ‘Number of Epoch’ and y-axis represents loss and can see with each increasing epoch loss values got decreased and reached closer to 0.



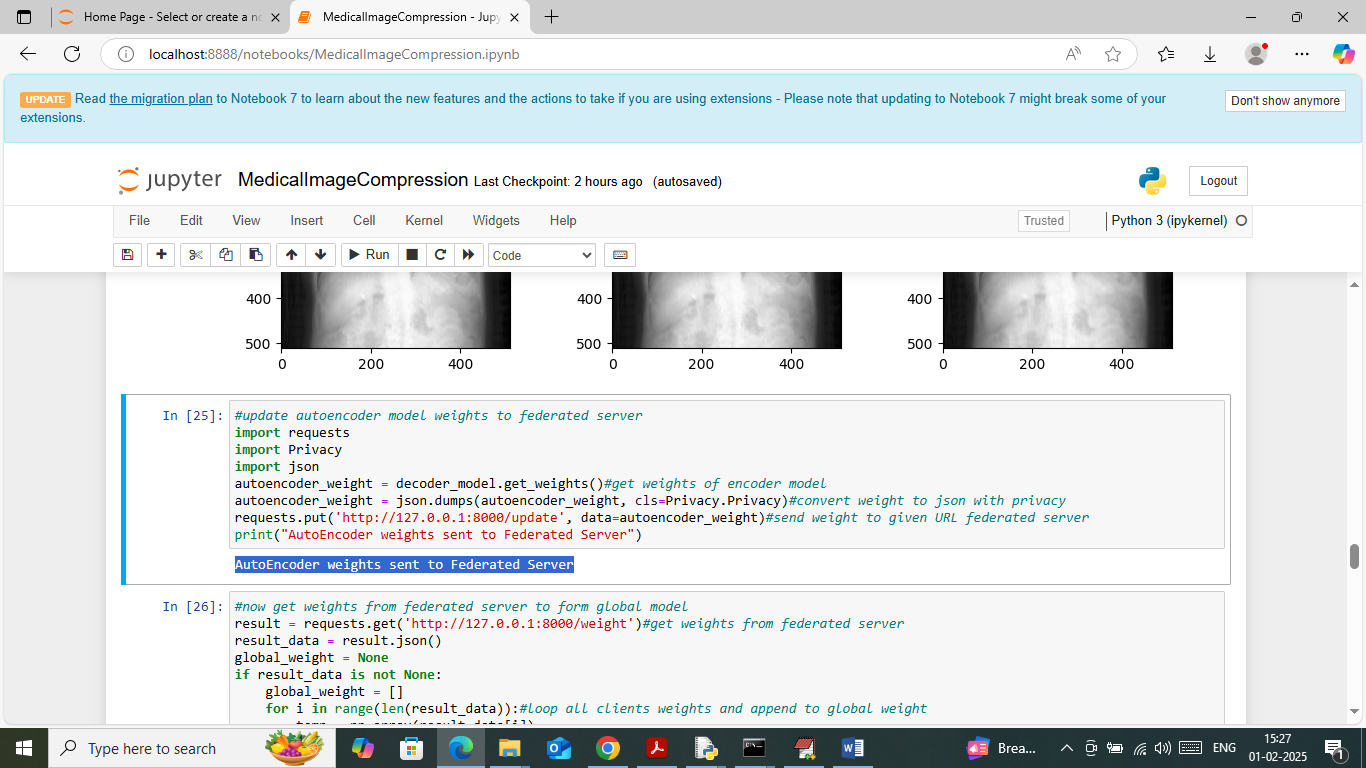
In above screen in tabular format can see original image size, JPEG2000 compress and OMMLSVD compress size and in all OMDSVL got less size



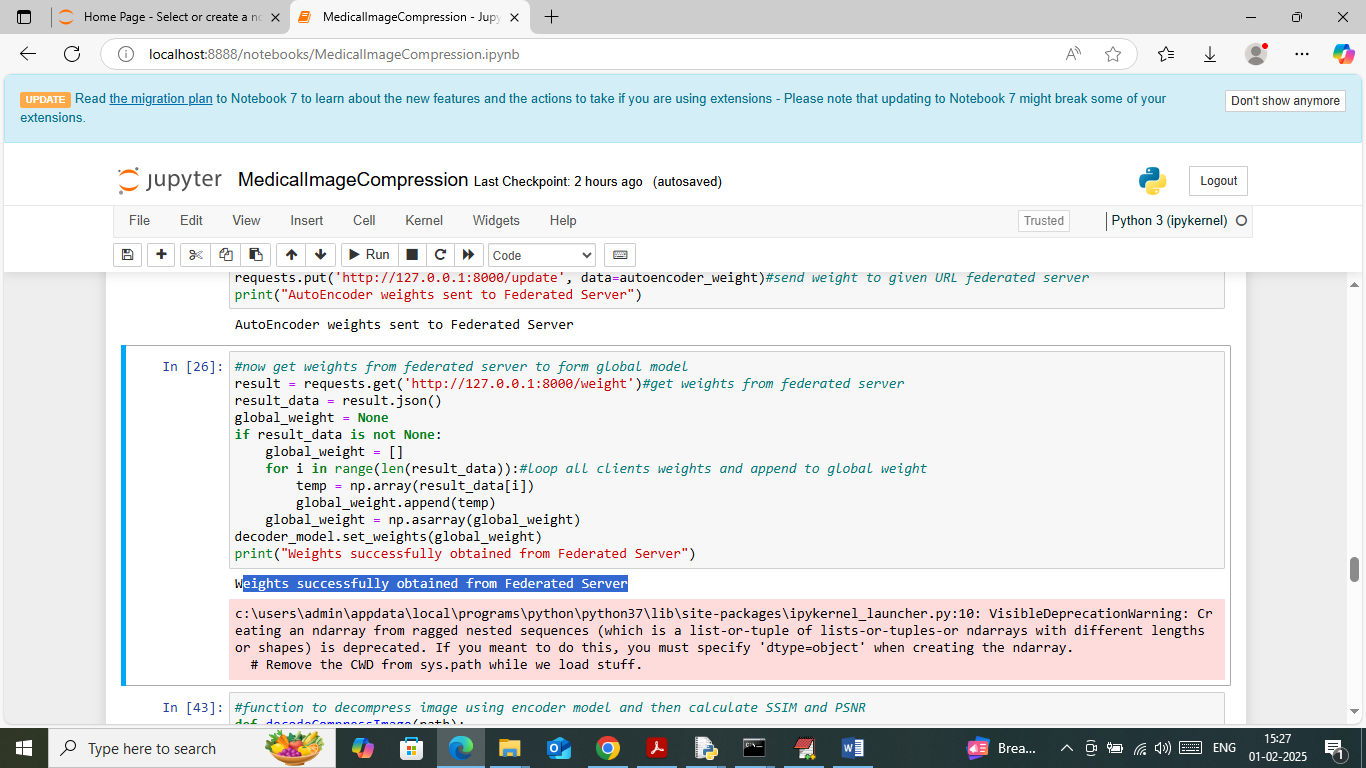
In above screen visualizing storage size comparison graph where x-axis represents compression type and y-axis represents size and in all propose OMLSVD got less storage size



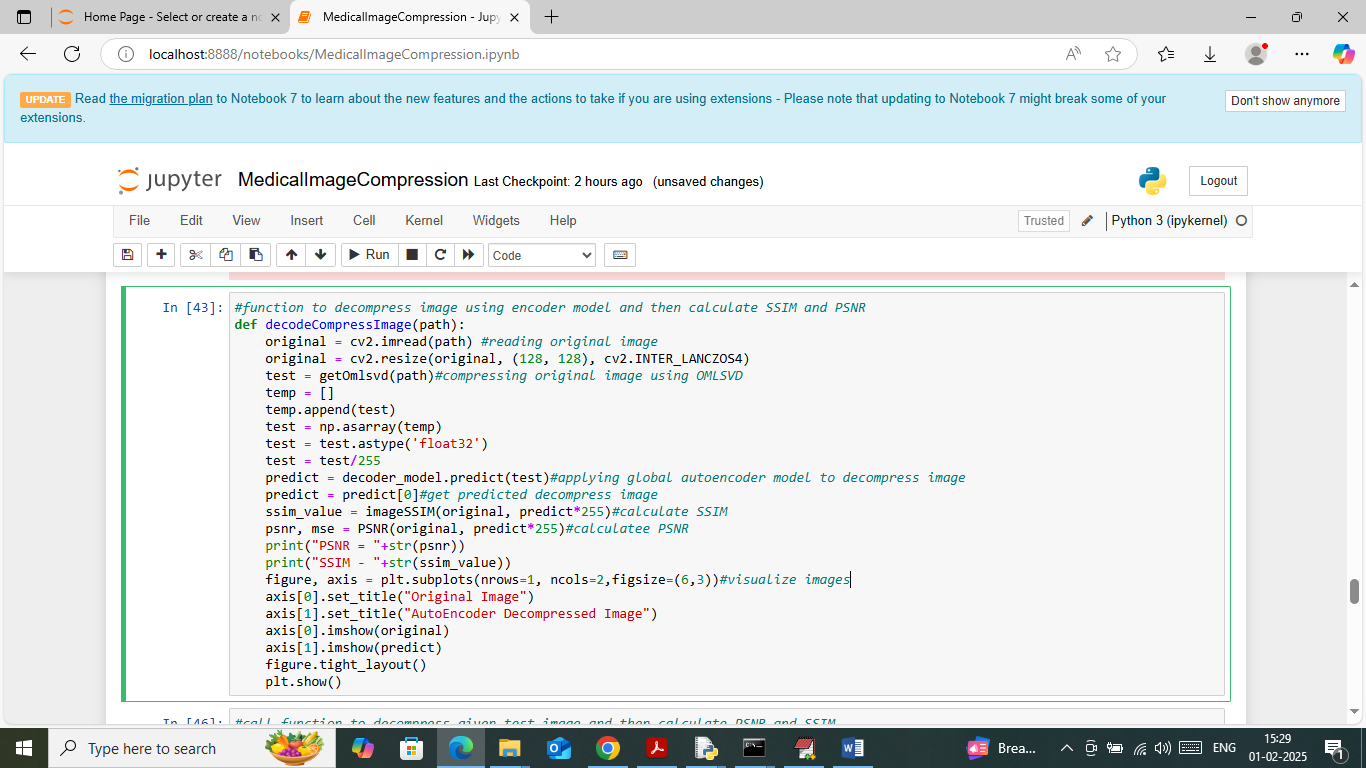
In above screen visualizing images where first image is the original image and second image is the JPEG2000 compress size and 3rd image is the OMLSVD compress image. In above screen can see all images look similar without any difference but we can see difference in storage size



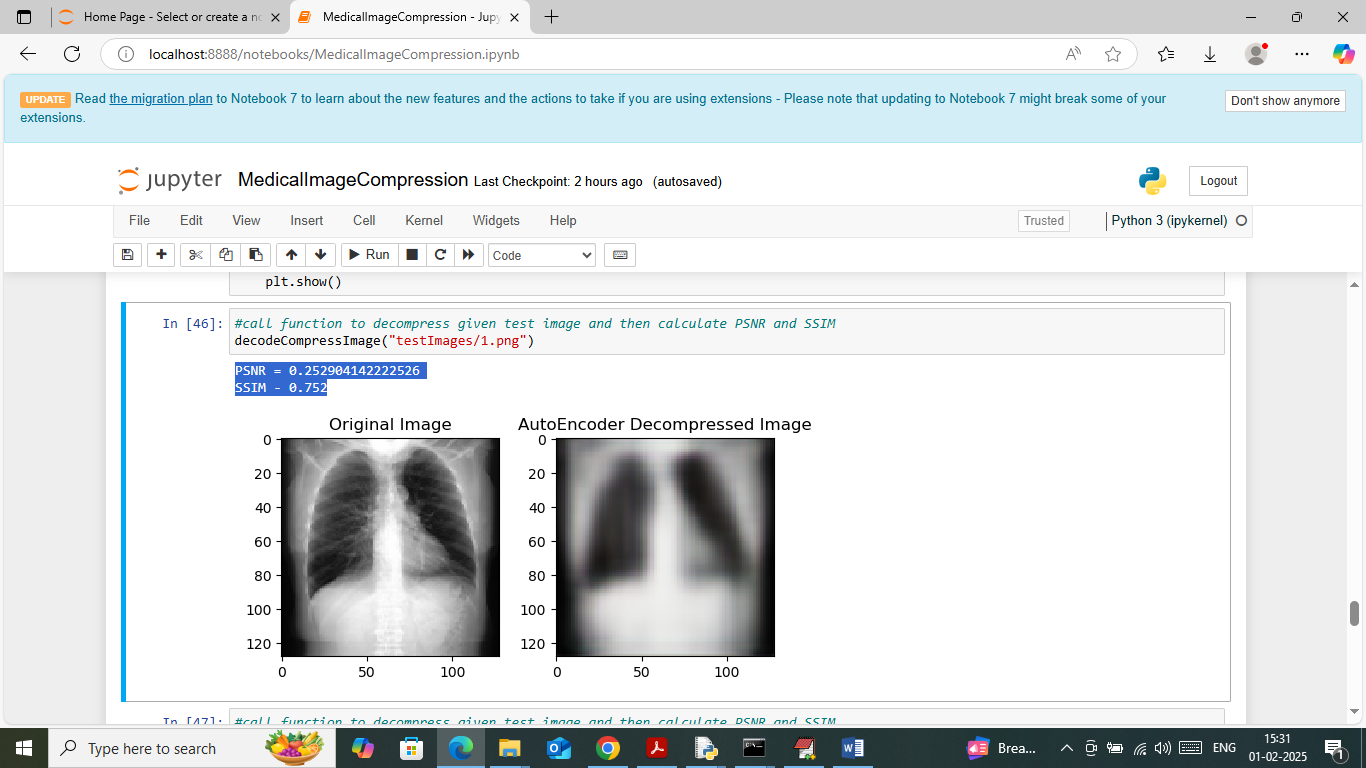
In above screen uploading auto-encoder model to federated server by using federated server URL



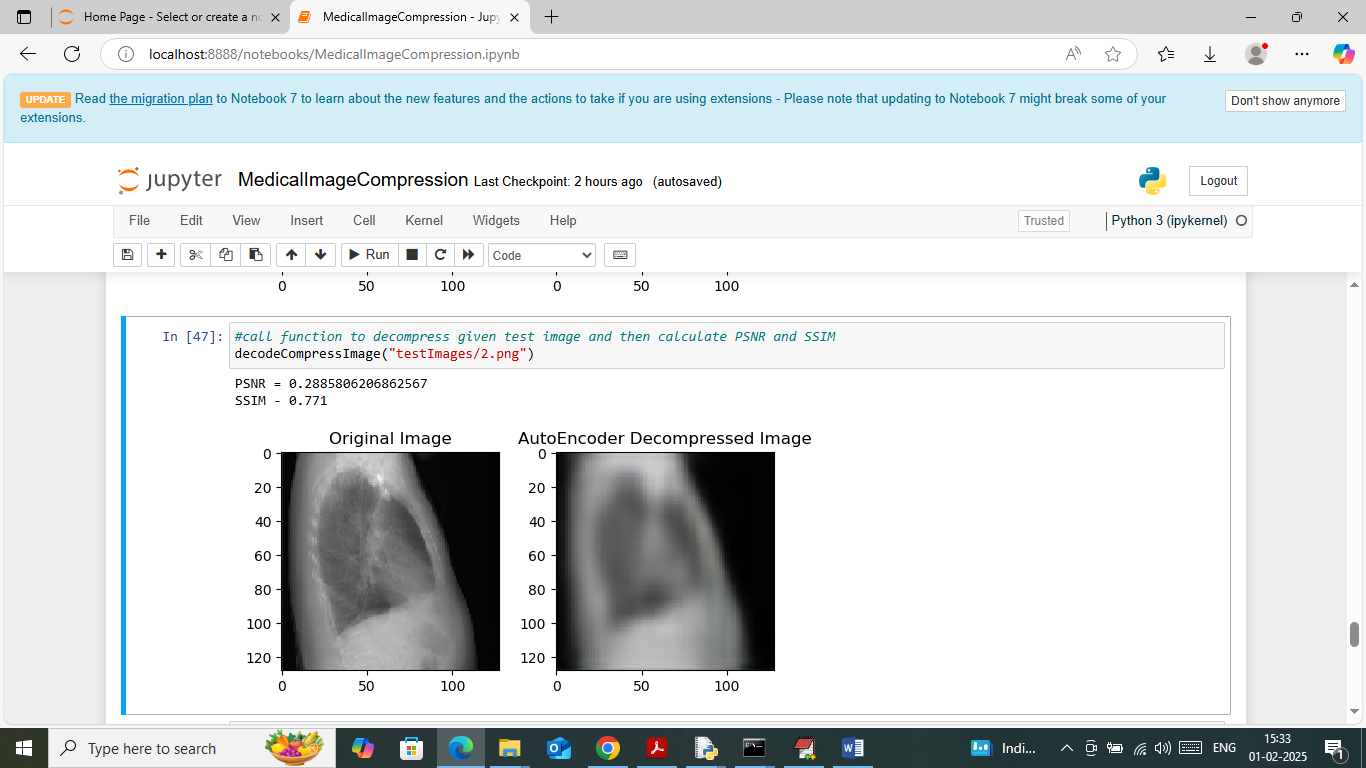
In above screen connecting to Federated Server and getting all weights as global weights



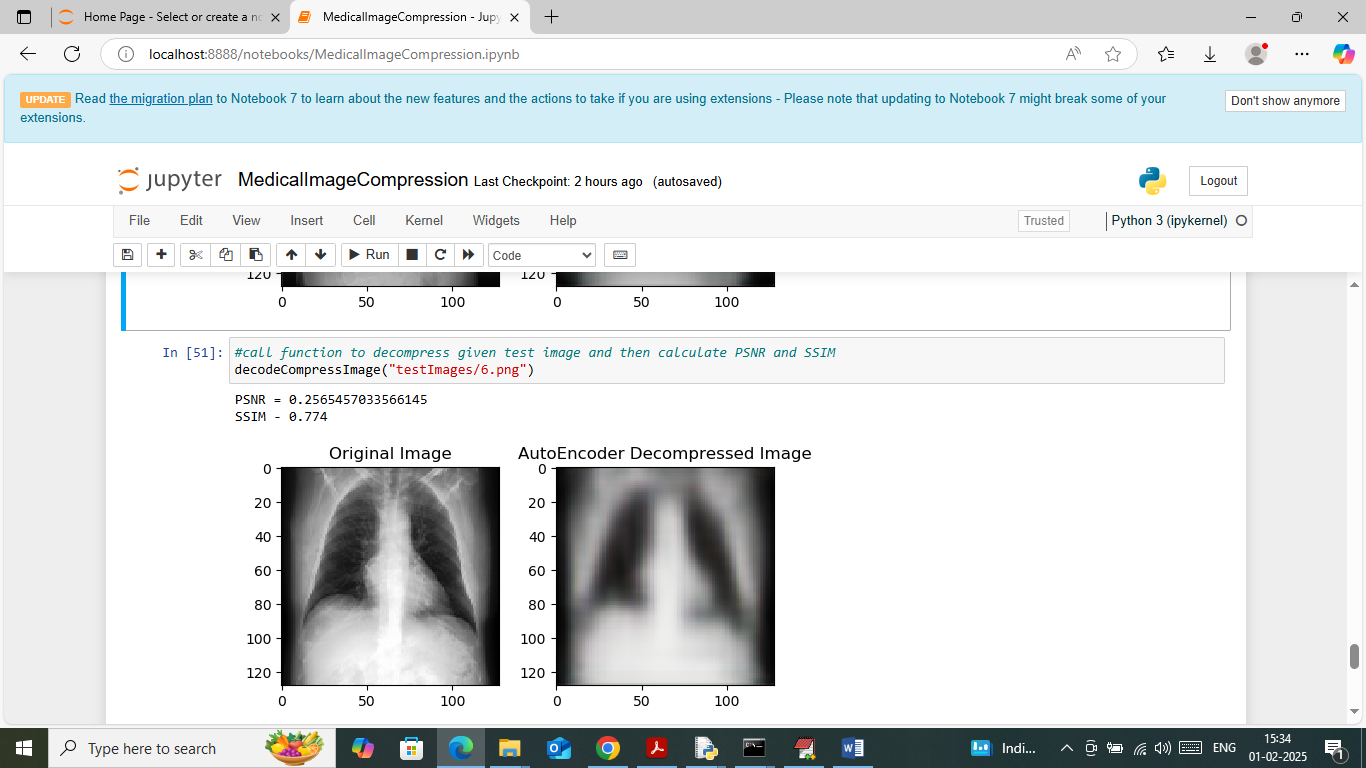
In above screen defining ‘Decoded decompress image’ function to dcompress image



In above screen calling decompress function with test image and then first image is the original image and second image is the auto-encoder decompress image which are looking same. In blue text can see SSIM and PSNR values where SSIM (structural similarity image) must be closer to 1 and PSNR (peak signal noise ratio) must be closer to 0.



In above screen predicting pr decompress another image out[ut where original and decompress images are same



In above screen can see another image output.

Similarly change test image path and predict decompress image for any other test image.