MDS Assignment 1

Chidaksh Ravuru(200010046)
Om Patil (200010036)
Hrishikesh Pable (200010037)

February 2023

1 Introduction

In question 1 of this assignment, we use SVD algorithm to compress images and videos, and store the relevant data i.e. the first k singular values, and the partial U and V matrices, which we then use to reconstruct the image and the video. Then we compare the newly obtained video or image using the mean squared error loss and the structural similarity score to compare the quality of compression for different values of k.

In the second question, we use the SVD algorithm to find the nearest movies to a given movie. Then we find a particular user's highest rated movie and recommend the top k most similar movies to that movie using cosine similarity.

2 Problem 1 (part A)

2.1 Solution

For compressing the image, first we seperate the image along its channels, and then for each channel, we use the SVD algorithm to get the U, S and V matrices. Now, we take the first k singular values from the set of all singular values i.e. we reduce the S matrix to a diagonal matrix consisting of the first k singular values. Then we similarly truncate U and V matrices to k in the first and the second dimension respectively. Then we multiply the three reduced U, S and V matrices, to get the compressed image along all 3 channels. Finally, we stack these 3 matrices to get the compressed image.

Then, we compare the compressed images with the actual image, by calculating the mean squared error and the structural similarity scores.

Following is an image that we compressed using SVD, and the different images that we got using the different number of singular values.

Figure 1: Original Image

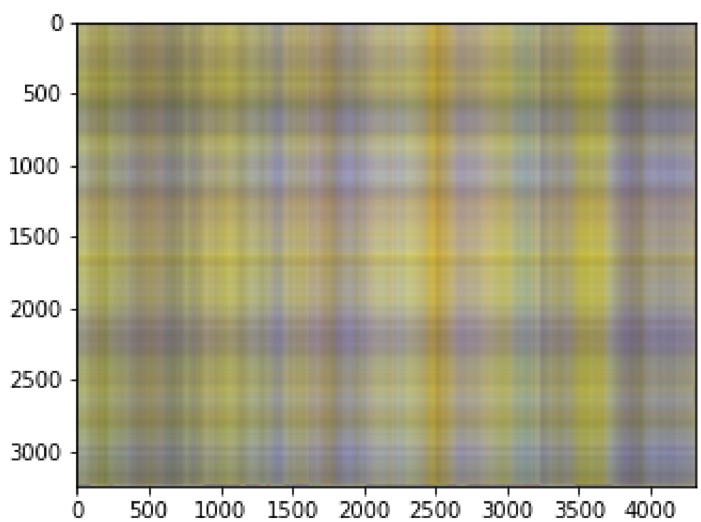

Figure 2: Compressed Image (1 singular value)

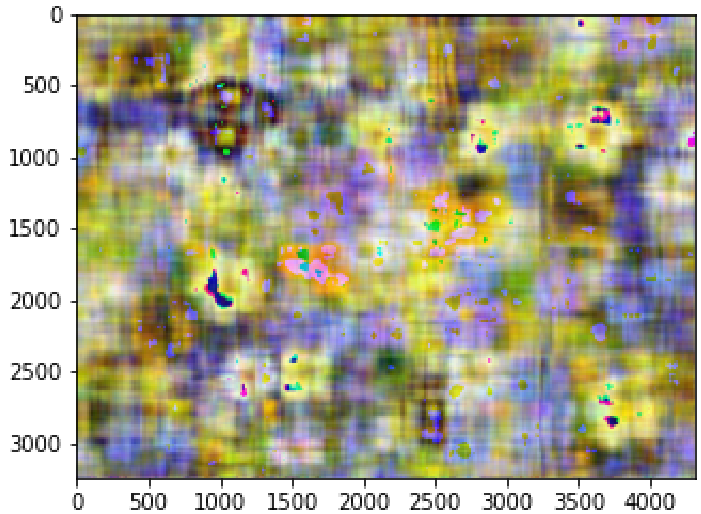

Figure 3: Compressed Image (11 singular value)

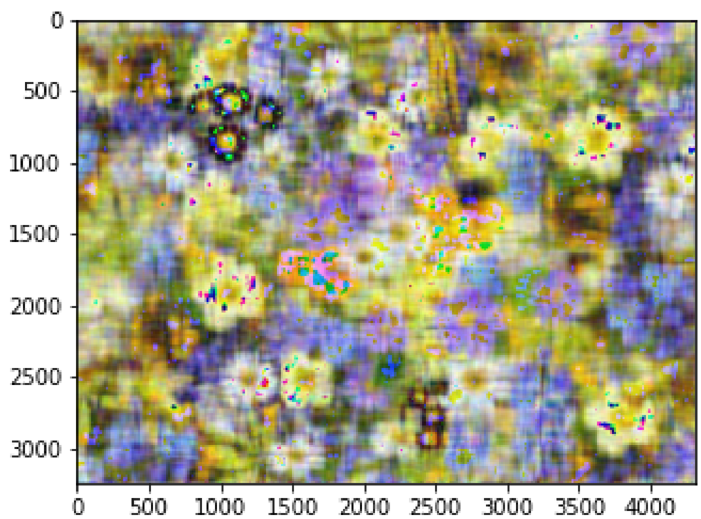

Figure 4: Compressed Image (21 singular value)

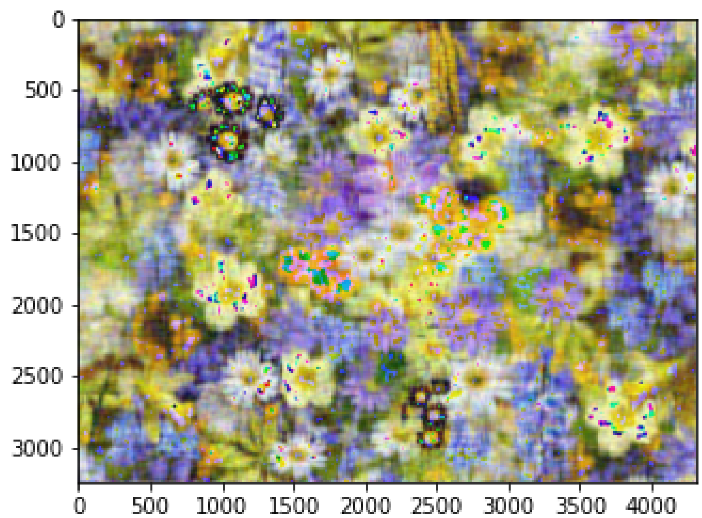

Figure 5: Compressed Image (31 singular value)

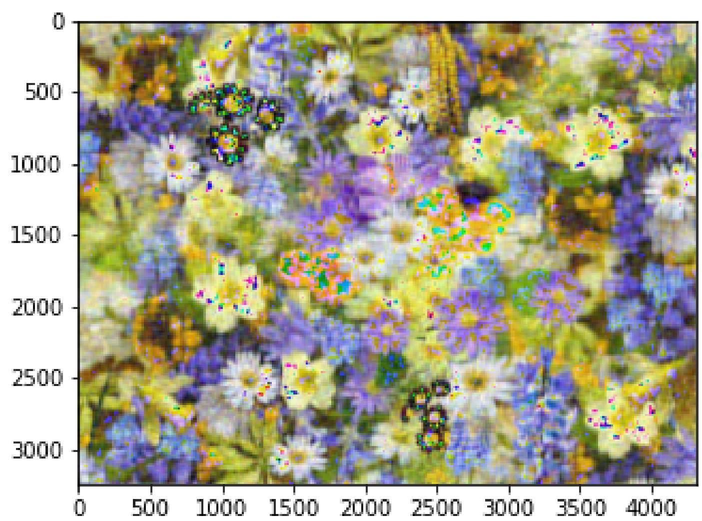

Figure 6: Compressed Image (41 singular value)

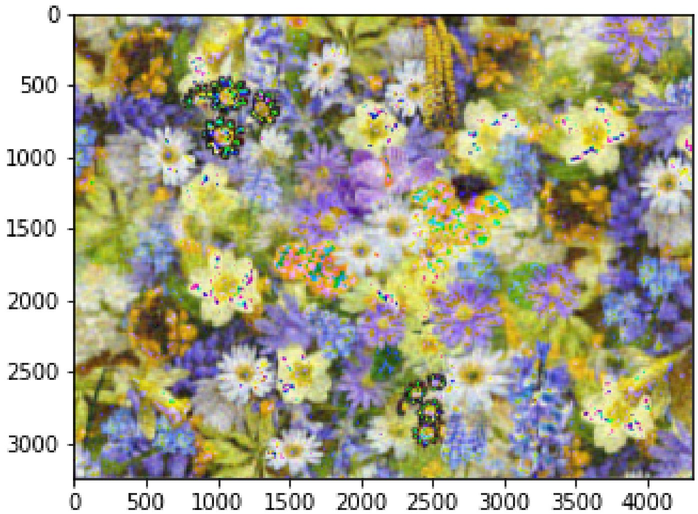

Figure 7: Compressed Image (51 singular value)

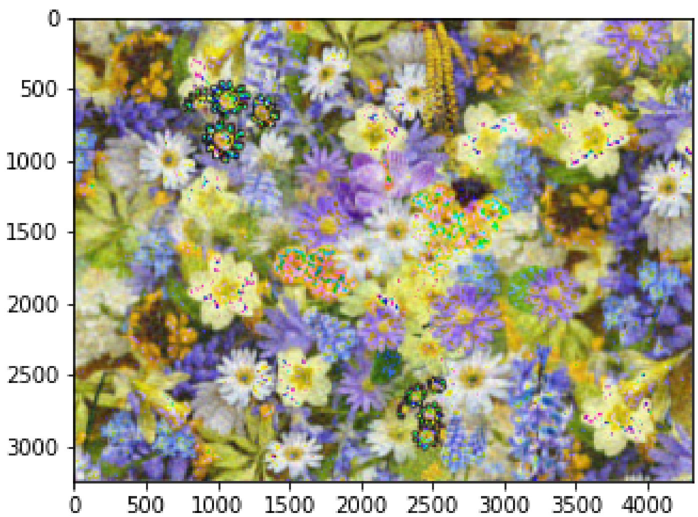

Figure 8: Compressed Image (61 singular value)

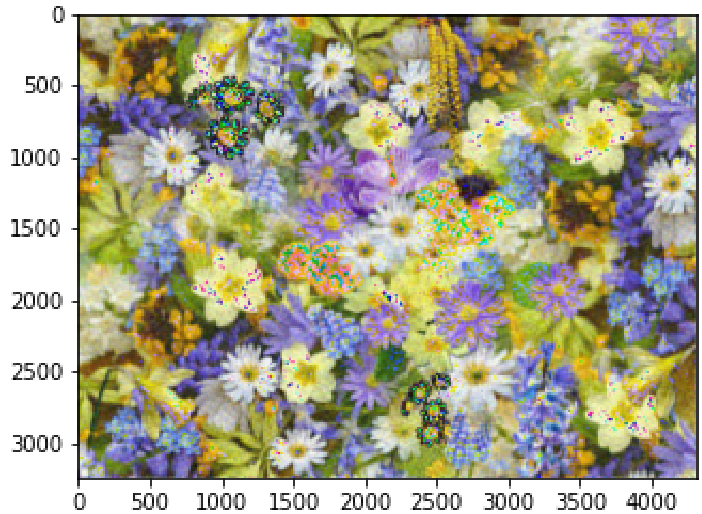

Figure 9: Compressed Image (71 singular value)

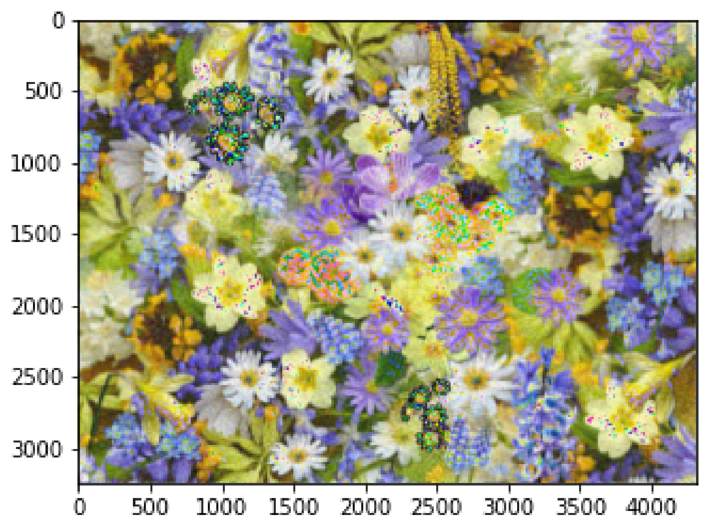

Figure 10: Compressed Image (81 singular value)

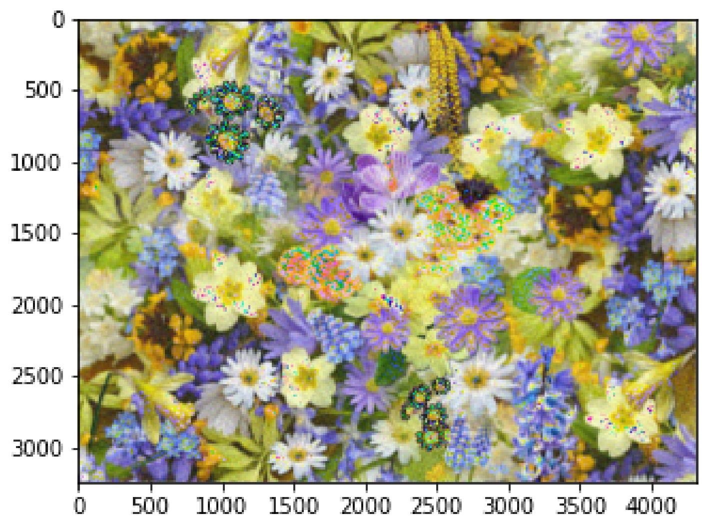

Figure 11: Compressed Image (91 singular value)

2.2 Observations

Following is the plot of MSE and structural similarity score against the number of singular values. As expected, the MSE decreases and the structural similarity score increases as we increase the number of singular values:

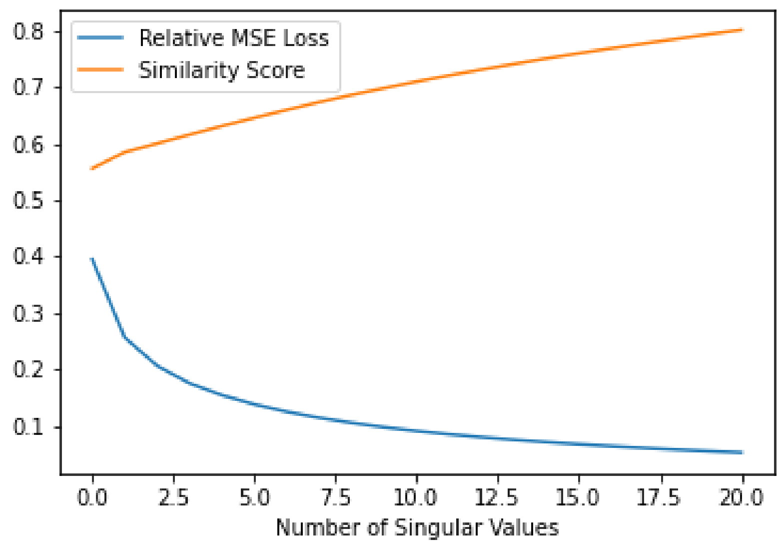

Figure 12: Caption

3 Problem 1 (part B)

3.1 Solution

For video compression too, we used SVD decomposition. We first save the number of frames, height and width of the frames in the video. Then we sequentially access the frames in the video and find the compresses image for that frame using the previous method. We store these compressed images into an array. After this, we create a video using these compresses image frames.

Now, in order to compare the original and compressed videos, we create a list of mse error values and the structural similarity scores for each of the corresponding images in the two videos. We take the average of the list and use that as the mse error and the structural similarity score for the video.

3.2 Observations

We ran our code on a 1 second long video having 30 frames. Following is the plot of mse and ssim for the video against the number of singular value:

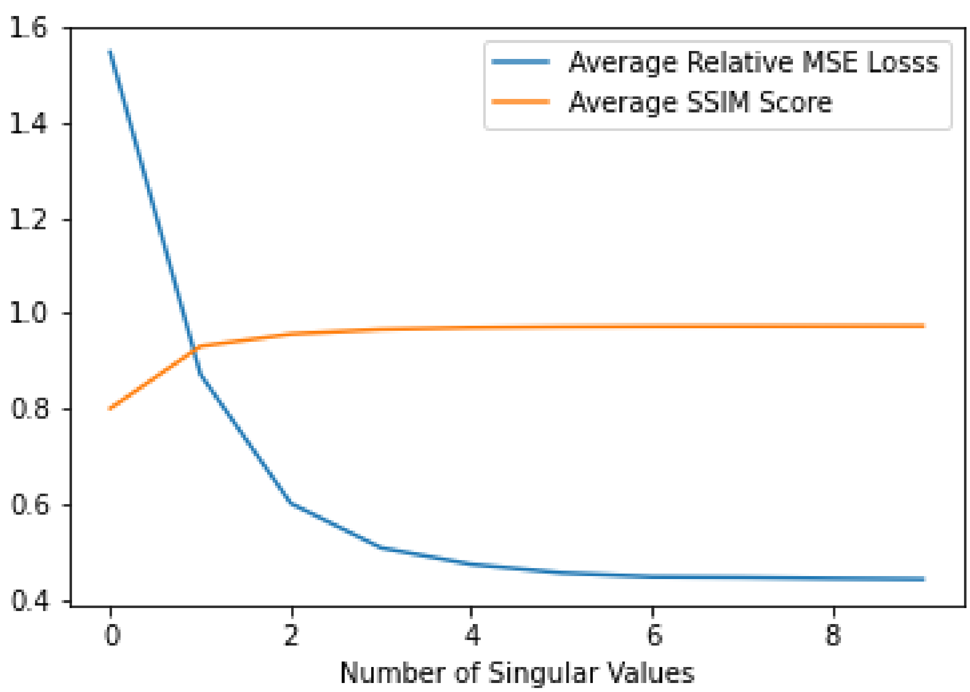

Figure 13: Caption

4 Problem 2

4.1 Solution

We first normalize the encodings of the movies according to the user ratings and also the encodings of the movies according to the genre, by subtracting the mean from each dimension. Then we use SVD decomposition over the normalised ratings and genre to get the latent features with lesser dimensions (50 and 10 dimensions respectively). Now, to find the top k similar movies for each given movie we use cosine similarity between these new encodings.

4.2 Observations

After we reduced the ratings and genre encodings of the movies to the latent space, with 50 and 10 dimensions, we further reduce them to 2 dimensional space for visualisation

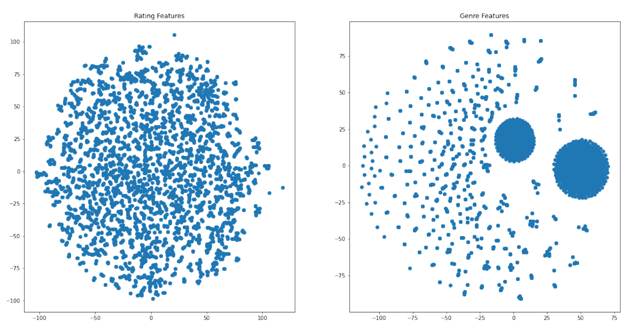

Figure 14: Ratings and Genre encodings