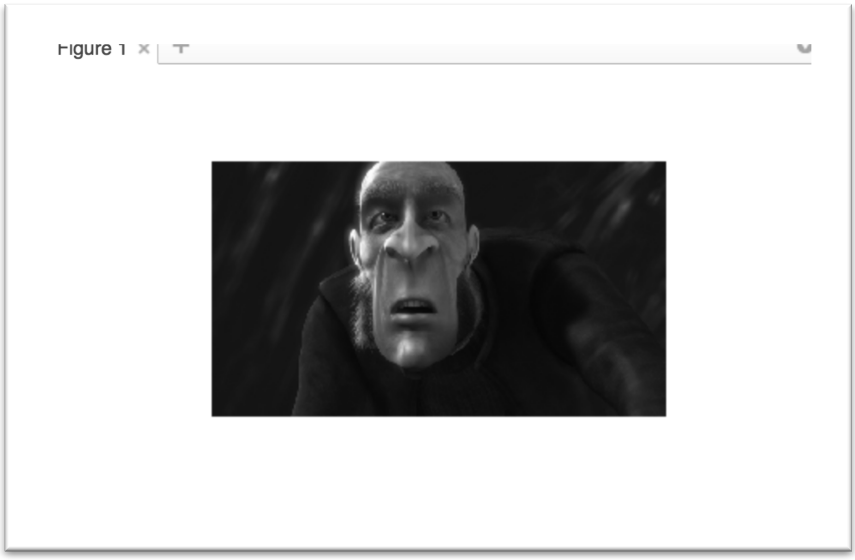
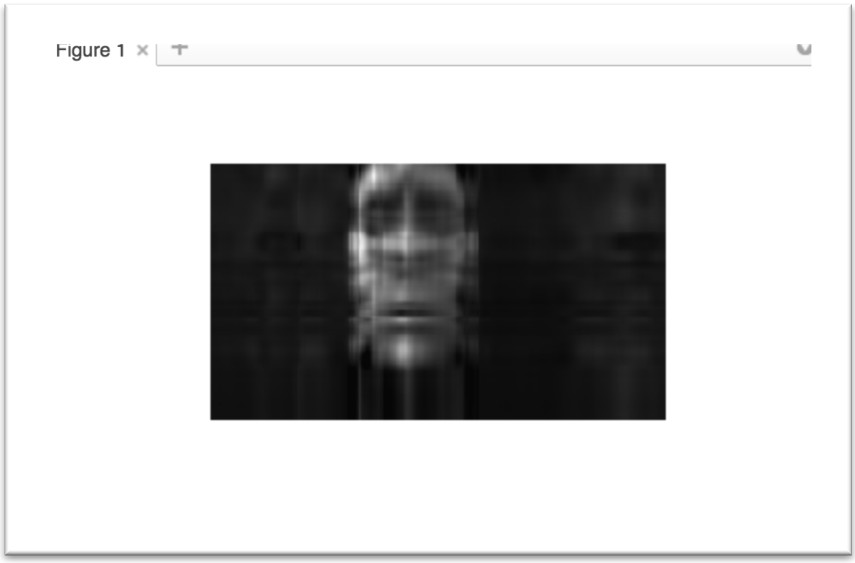
Non-negative Matrix Factorization Experiment 1 with Images from Elephants Dream

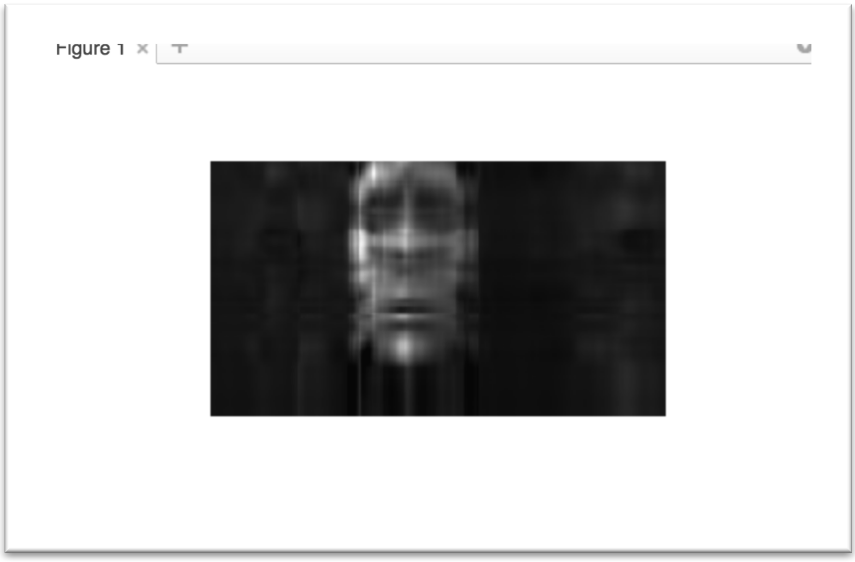
July 25, 2017



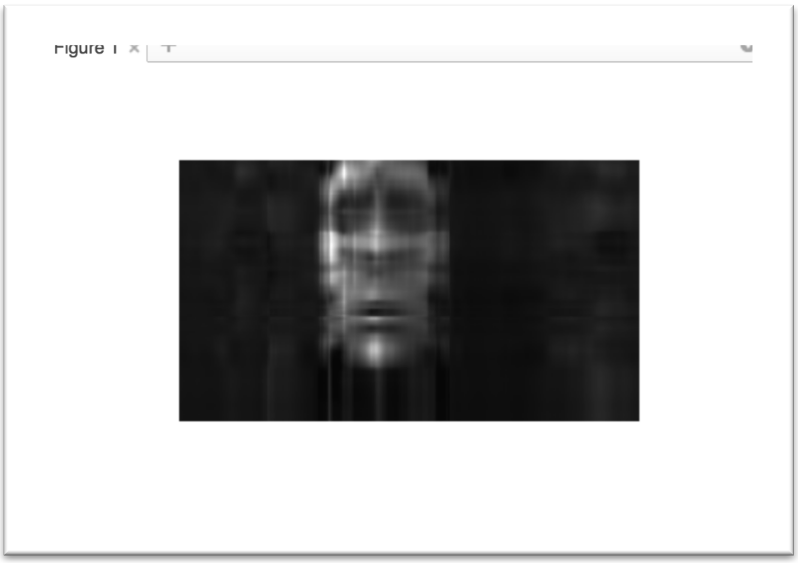
(Fig. 1) Original image



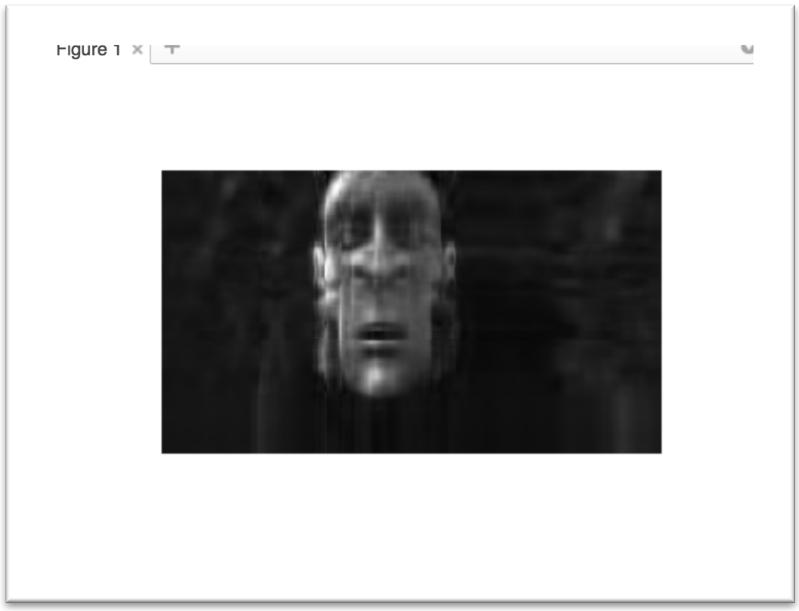
(Fig. 2) Predicted Image 1 (**d_cols = 4 & NumIter = 20)**
14.6892 norm difference



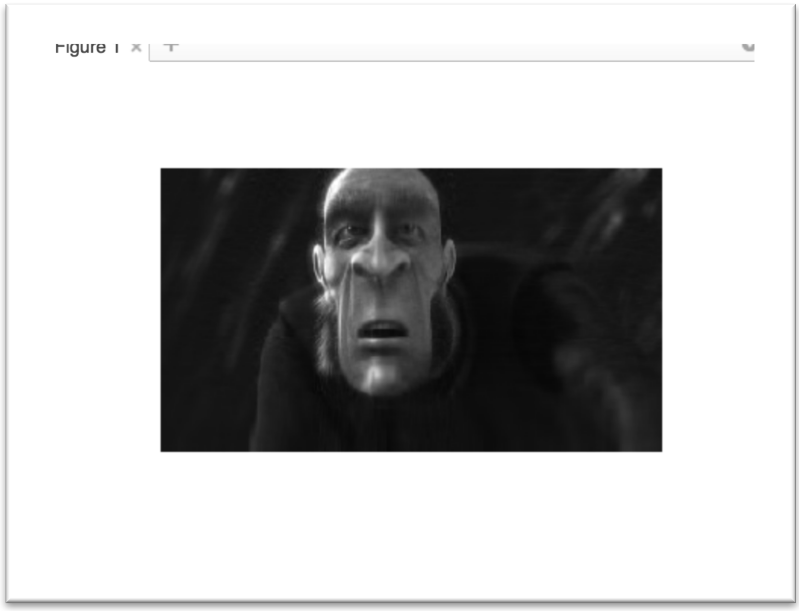
(Fig. 3) Predicted Image 2 (**d_cols = 4 & NumIter = 50)**
14.5717 norm difference



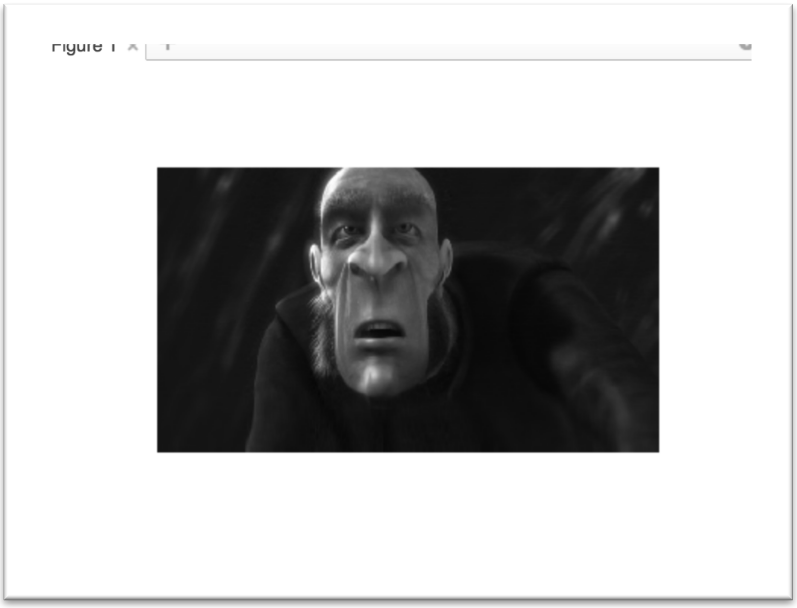
(Fig. 4) Predicted Image 3 (**d_cols = 4 & NumIter = 100)**
14.6554 norm difference



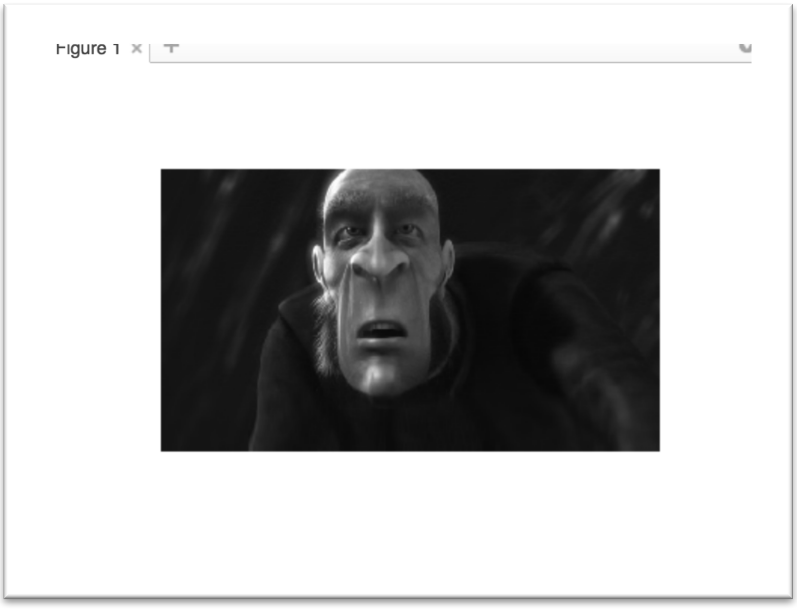
(Fig. 5) Predicted Image 4 (**d_cols = 10 & NumIter = 20)**
5.8890 norm difference



(Fig. 6) Predicted Image 5 (**d_cols = 50 & NumIter = 20)**
1.4912 norm difference



(Fig. 7) Predicted Image 6 (**d_cols = 100 & NumIter = 20)**
0.6349 norm difference



(Fig. 8) Predicted Image 7 (**d_cols = 100 & NumIter = 500)**
0.6482 norm difference

CONCLUSION:

After I have gathered images from the Elephants Dream website, I imported them into our Matlab program. The matrices created based on the images were initially of size 360x600x3; however, using **rgb2gray** function, I was able to convert these "3D" matrices into 2D ones with no RGB values.

Having done that, I began to play with the different parameters in order to see what makes the algorithm worse or better in terms of finding **X** and **Y** matrices that makes true "**Amat = XY"**—where **Amat** being the matrix generated based on the imported image.

1) Keeping the **d_cols** parameter fixed, I increased the **number of iterations** a couple of times and observed that it did not have significant influence over the predicted images—as is shown in the figures above.
2) After that experiment, I started manipulating the **d_cols** parameter as I kept the **number of iterations** fixed. I have seen that the **d_cols** parameter's value—which is the number of columns in **X** and the number of rows in **Y**—has a great effect on the predicted images. As can be seen above, while **d_cols = 4** did not give us a satisfying **predicted (X*Y)** matrix and a good image, **d_cols = 100** brought us much better **X** and **Y** matrices—which eventually results in a better **predicted** matrix that is very close to our original **Amat.**
3) In conclusion, it seems that the **d_cols** parameter's value has a very significant impact on the accuracy—or closeness to **Amat**—of the predicted matrix. Having said that, it is also important **not** to have a very large **d_cols** value, as it slows down the process significantly.