Image processing and analysis 00460200

HW 3 Wet, due to 13/07/2025

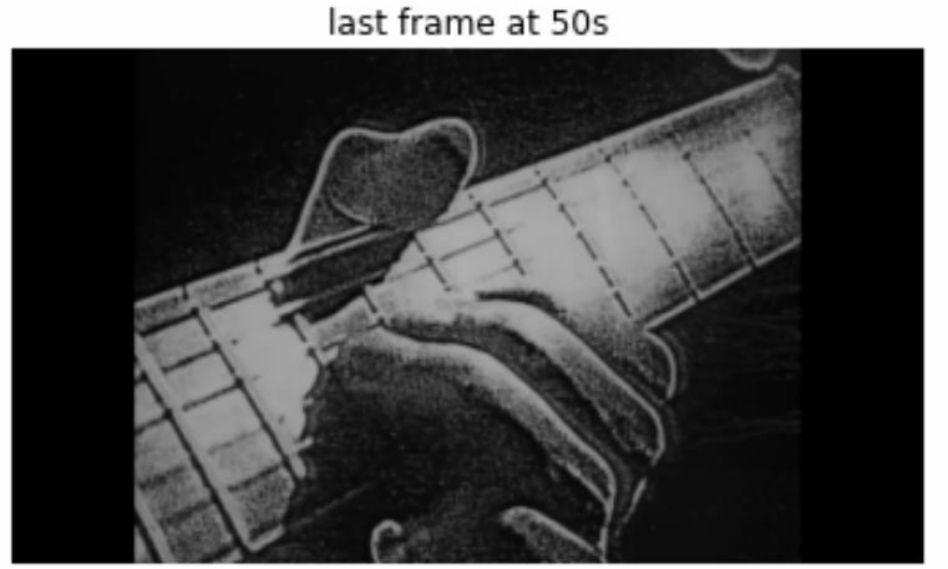
Lavie Lederman 319046504

Shahar Yankelovich 206159527

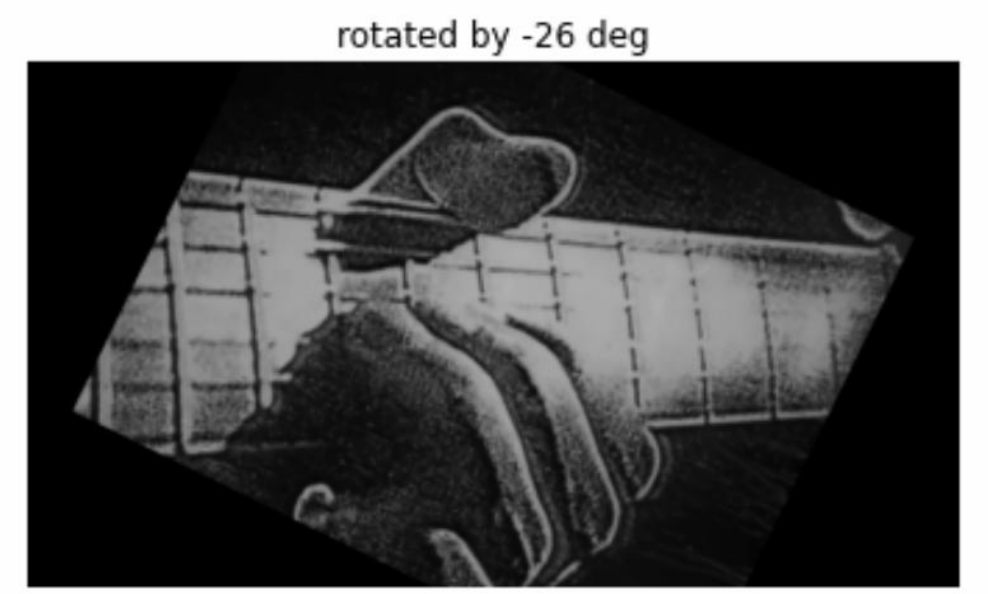
Part 1

1.a

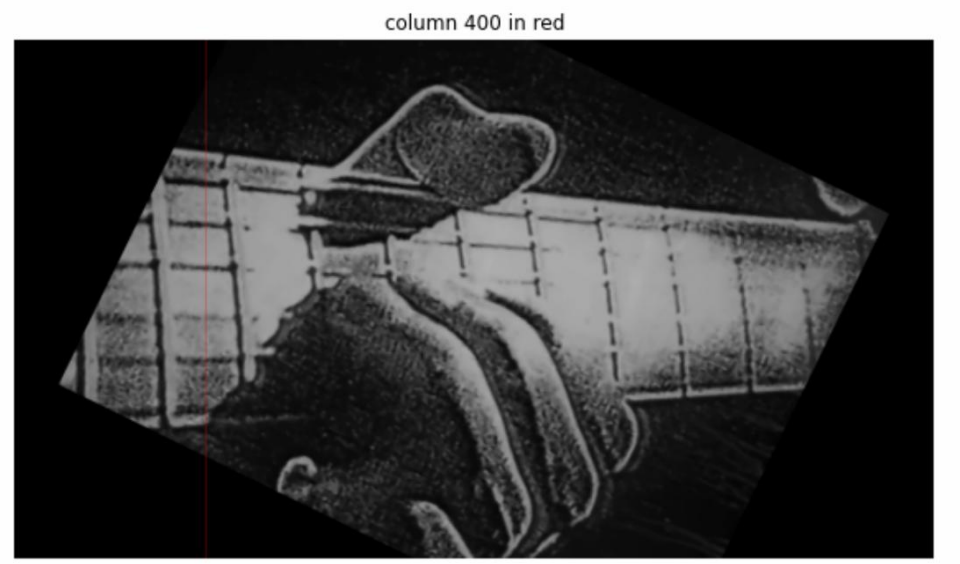
Here is the wanted frame from the 50th second –



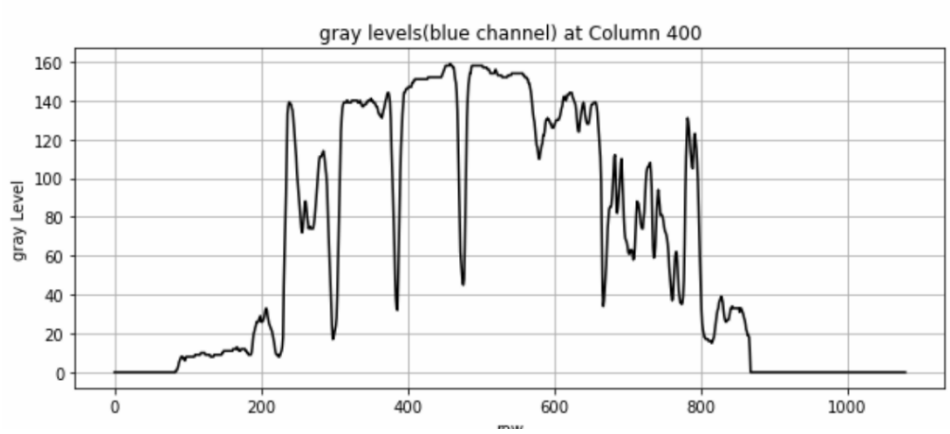
Here is the rotated image (we rotated it by $-26°$) –



Marked in red is the wanted column –

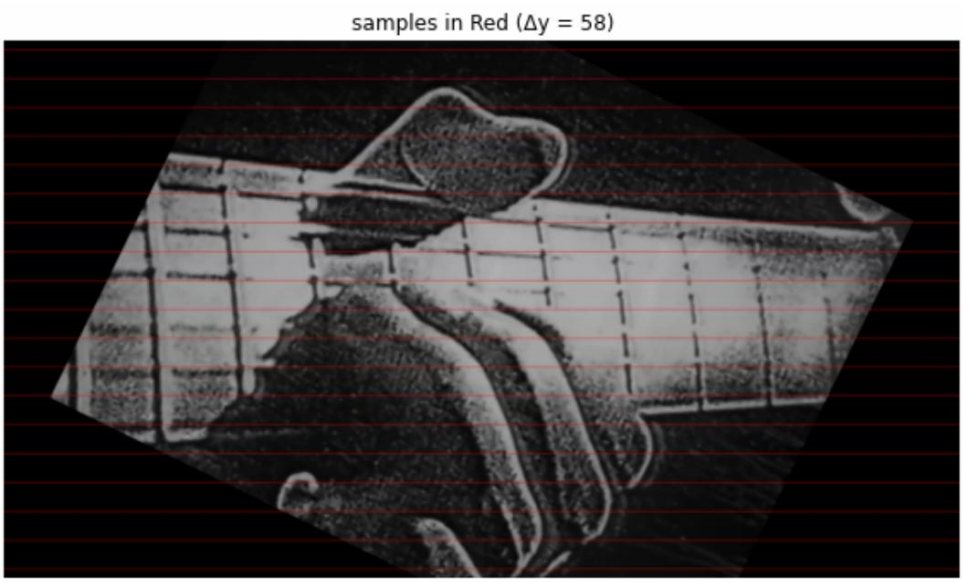


This graph shows the grey levels of the **blue** channel as a function of the row index –



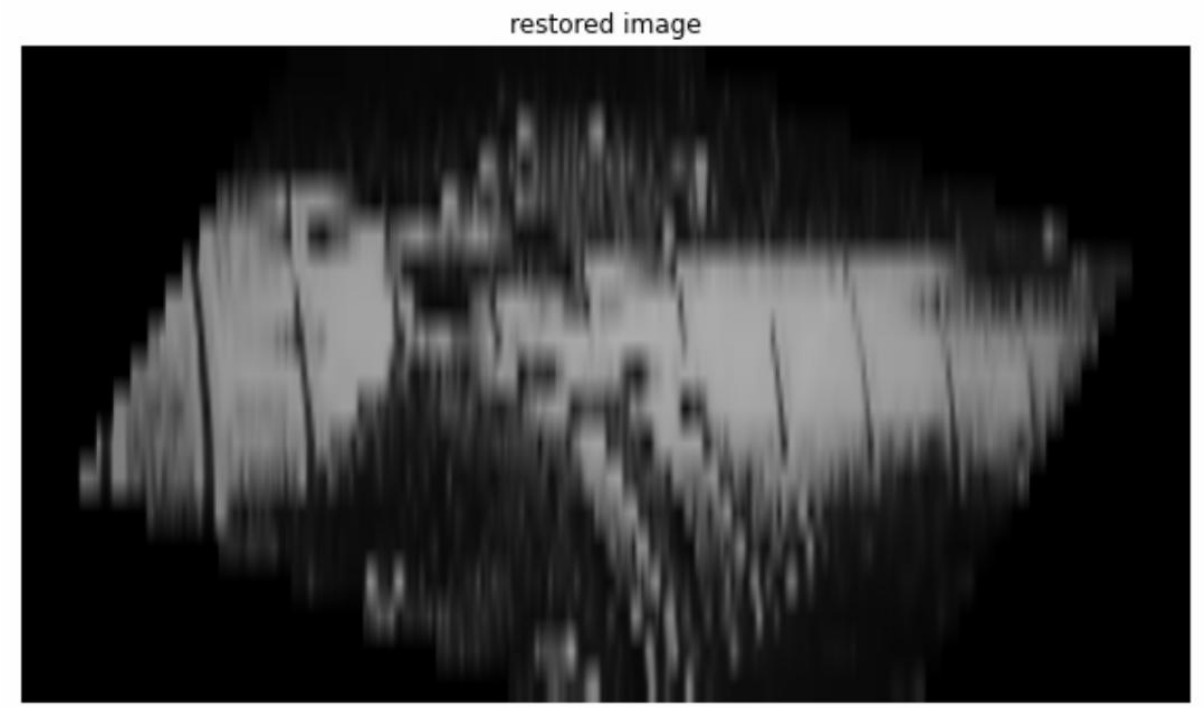
We can see that there are 6 shadows, some of them are easy to determine and others are less significant but from the periodicity and our prior knowledge of guitar structure we can conclude there should be shadows (and indeed there are patterns of shadow).

Ther sampled image is this –

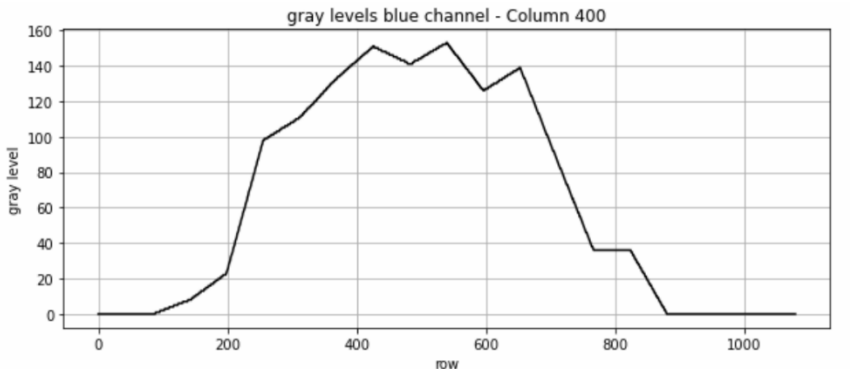


These are the rows that were sampled –

The result of returning the image to its original size is this –



We can see from the reconstructed image that there are no significant shadows. It makes sense, because the sampling rate in the vertical dimension is very low, and the shadows (that represent strings) are very thin and easy to miss. This is caused due to aliasing – the sampling rate is much lower than the spatial frequency. On the other hand, we can see that horizontal information is maintained to a level that allows us to identify patterns in this direction like fingers and vertical lines on the guitar's bridge.



As we can see, here there are no significant shadows in the vertical dimension. This graph is much smoother than the previous one – and that is a result of interpolating by the function.

1.b

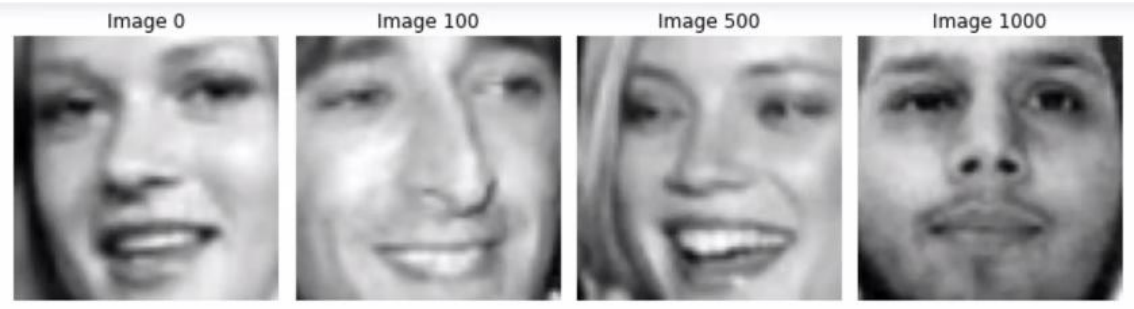
After watching the video, we can determine that the period time is approximately 3 seconds – meaning the frequency is about 0.33 Hz.

The video with 37 frame jumps shows the pendulum on either the extreme right or left of its swing, switching every sampled frame, we could double the jumps(to 74 for example) to skip over the frames on one side and be left with a video of the pendulum staying in the same place(we tried it with 74 and it works well enough, the pendulum moves a bit but stays around the same corner).

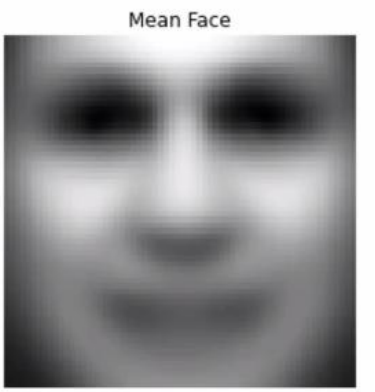
Part 2

2.a

Here are 4 images from the given array –

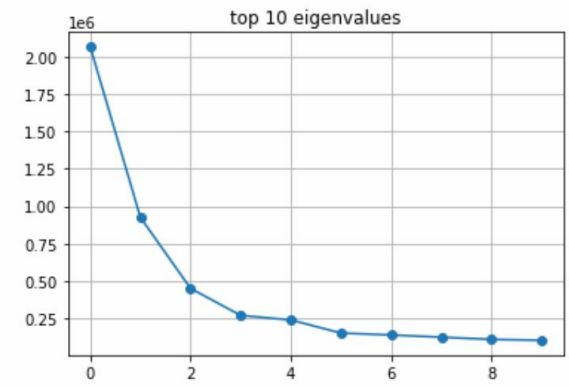


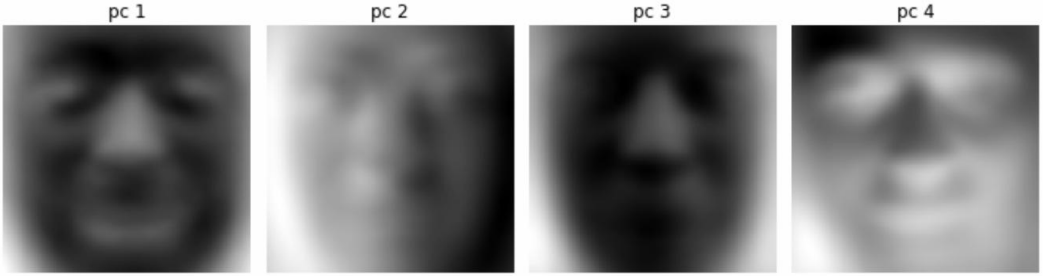
The following image shows the mean of every pixel –



2.b

Here are the values of eig_vals and first 4 principal components –



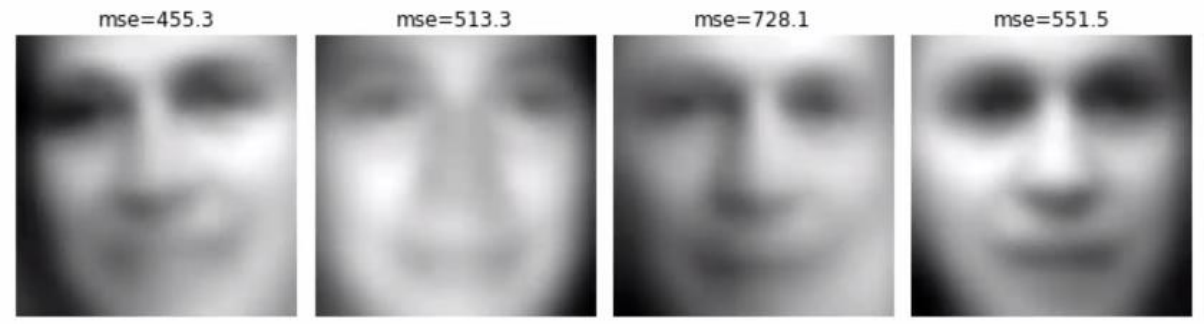


2.c

Done in the code.

2.d

Here are the restored images and their MSE –



2.e

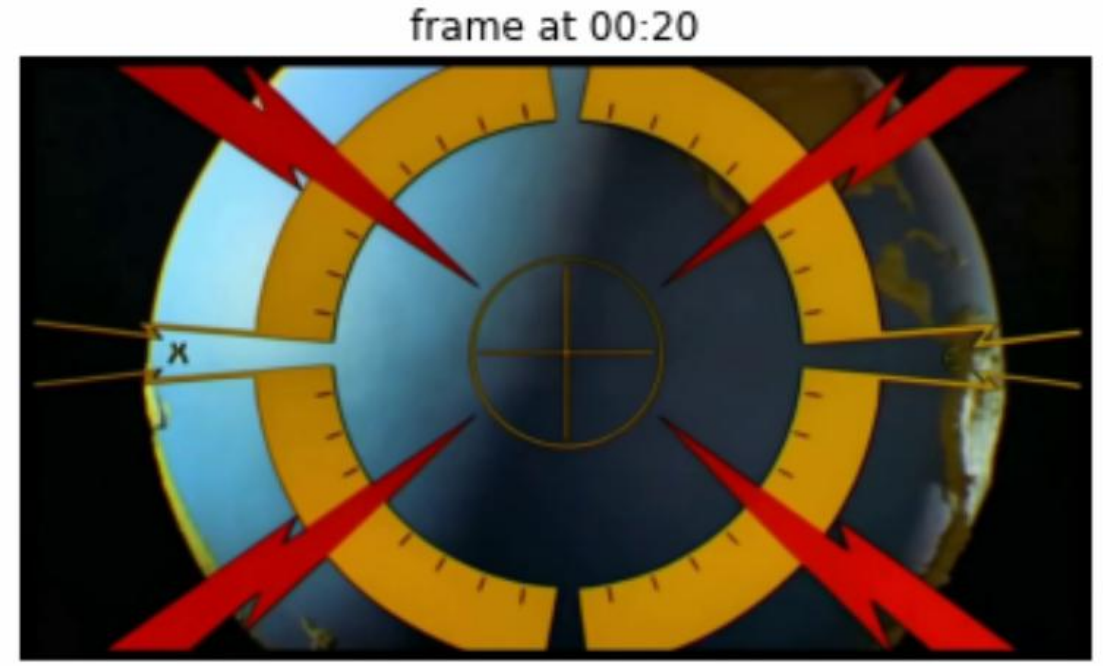
Now the results with k=570 –

As we expected the last results are much better with lower MSE and clearer image – this is a result of considering more eigenvalues – and by that maintaining more information. The tradeoff in this case is the memory usage which is bigger than the previous case.

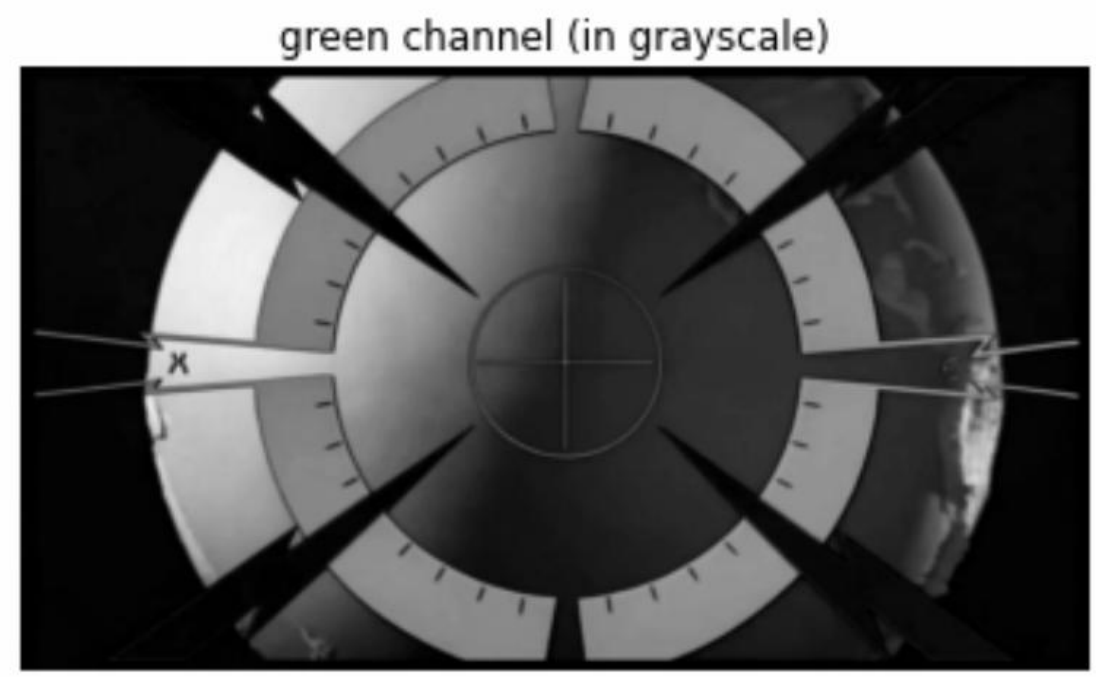
Part 3

3.a

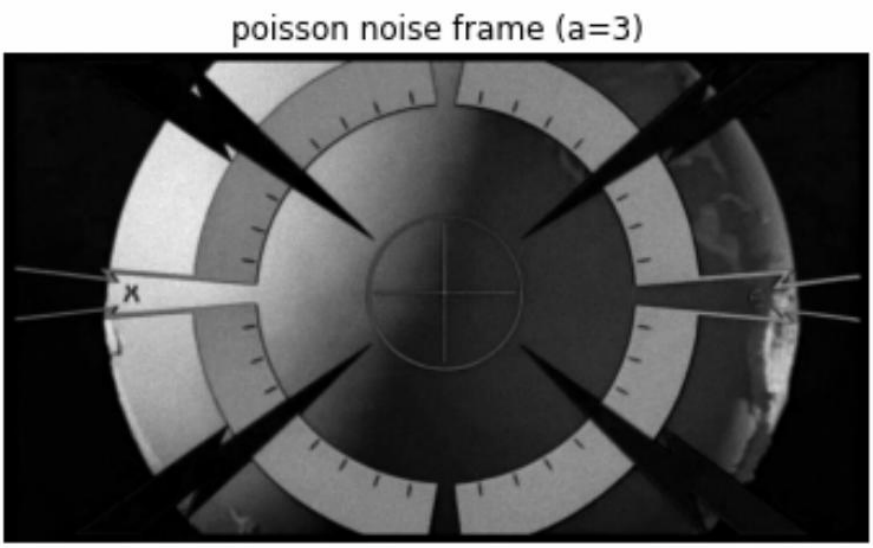
Here is a color image from the time section –



The grayscale of the green channel –

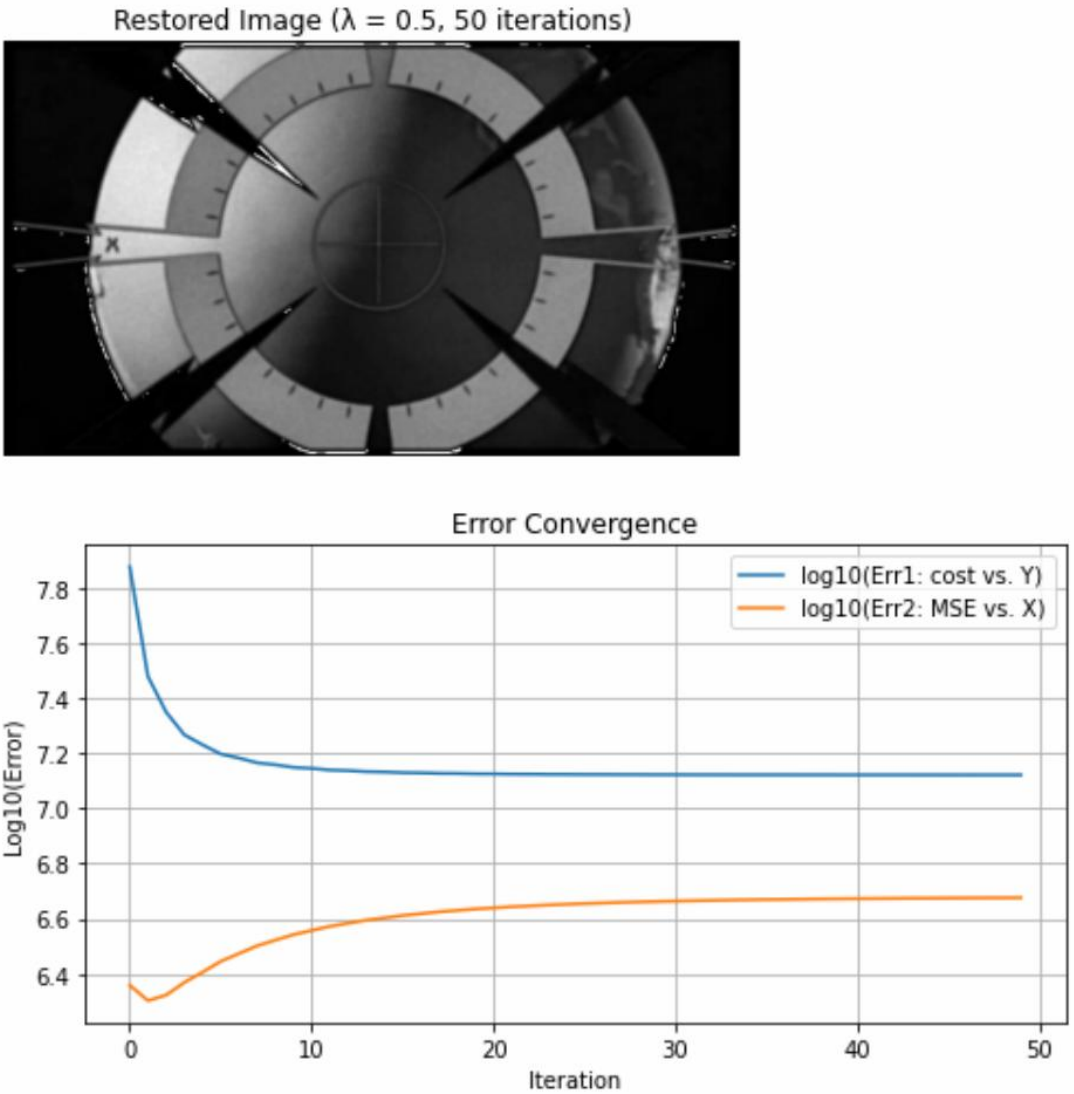


The noisy image –



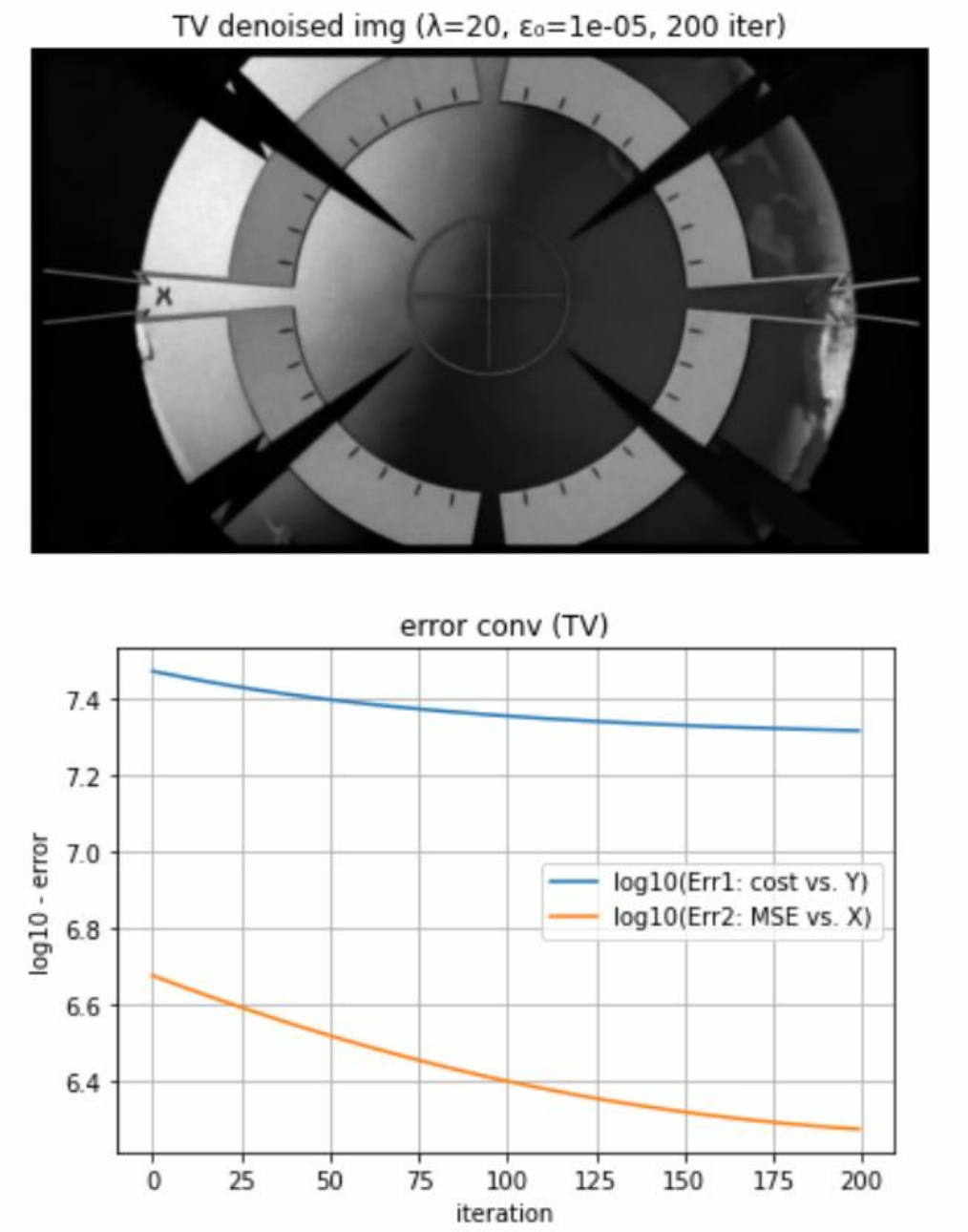
3.b

Here is the image restored with L2 and the logarithmic graph of Err1 and Err2 –



3.c

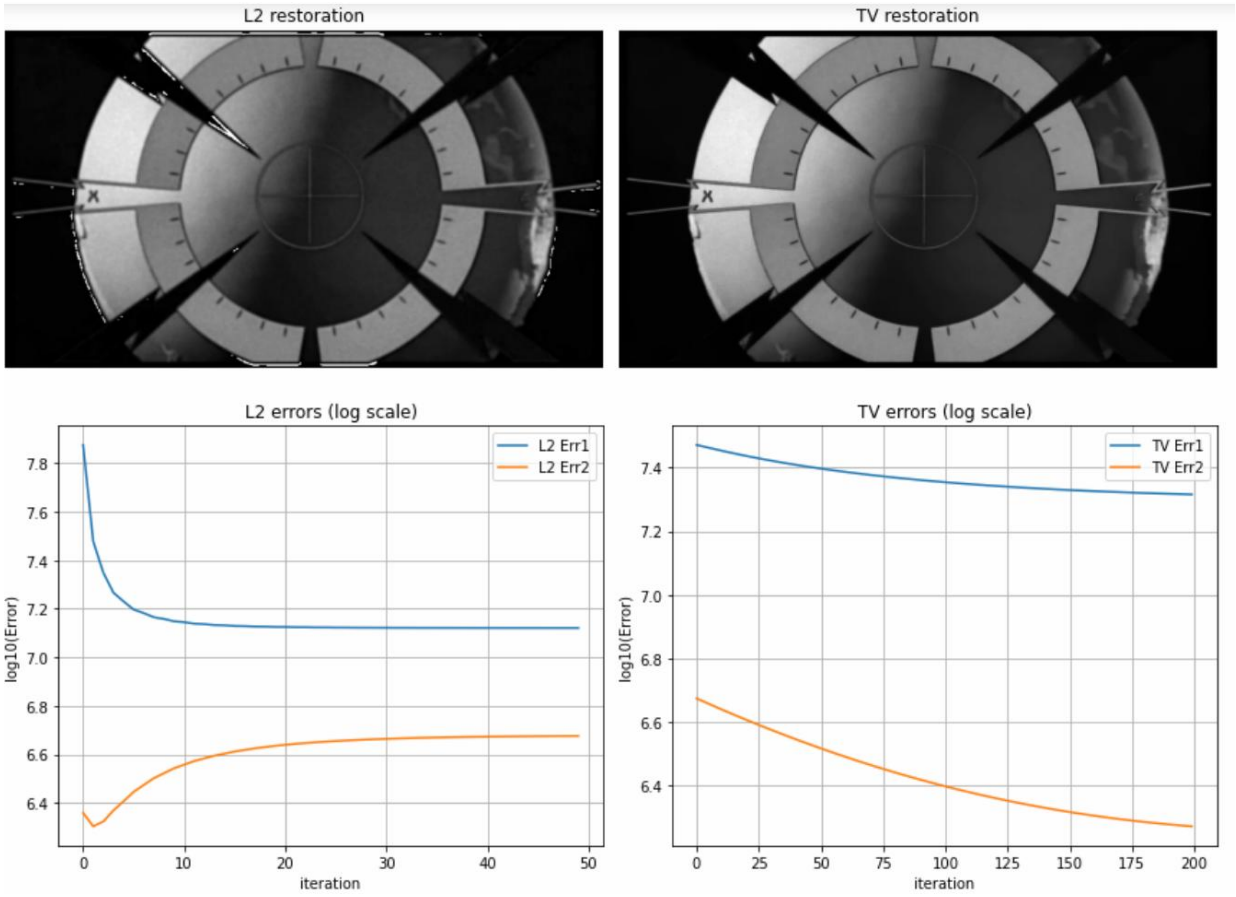
Here is the image restored with TV and the logarithmic graph of the new Err1 and Err2 –



We chose $\varepsilon_0 = 10^{-5}$ since it shows the best convergence and lowest value as we can see in the image above.

3.d

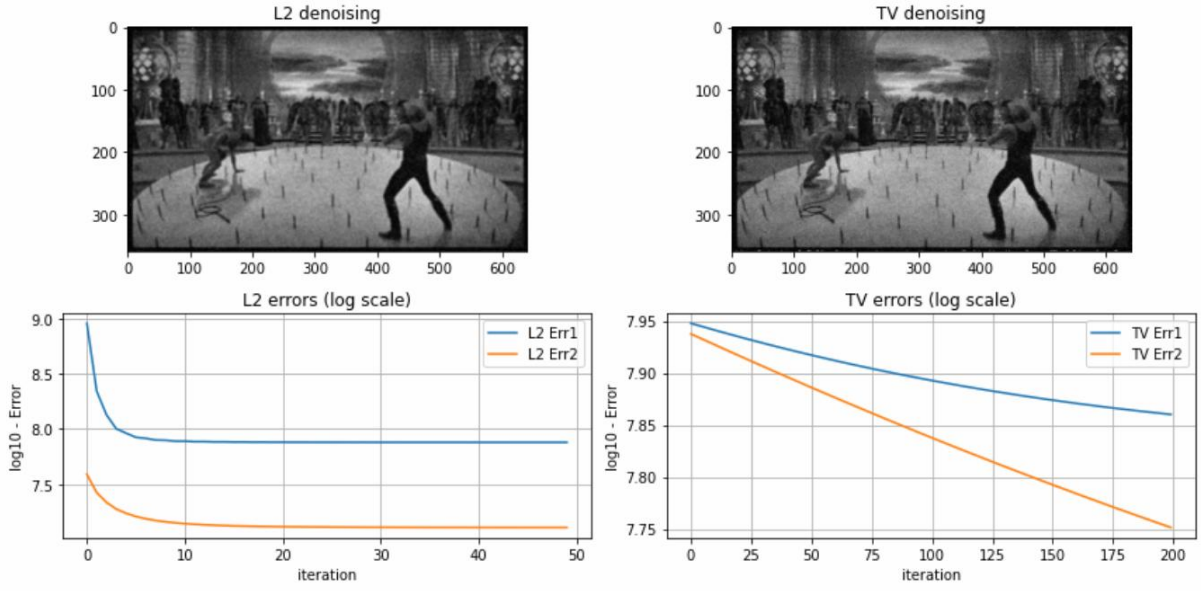
Here are the two results side by side –



As we can see, TV performed better since it promotes piecewise smoothness preserving edges and removing noise. On the other hand, l2 regularization tends to blur out both details and noise – which resulted in poor image in our case. Nevertheless, l2 showed faster convergence, meaning it saves resources. The choice between them is a tradeoff between performance and price.

3.e

Here is the restored image using each method on the frame at second 38 and their errors –



For natural images, we expect l2 to perform better since natural images tend to be smoother – and l2 denoising performs better on smooth images, as explained above.