2021

Image Quality Assessment on JPEG Compression Quality

GROUP 30
JOSEPH CUSHMORE AND ERIC MUNOZ

Problem Statement:

In this paper, we will use image quality assessment metrics to analyze image degradation between two JPEG images of the same scene. This assessment will be done over multiple generations, by using JPEG lossy compression on an original reference image. The goal of this experiment is to simulate image degradation over an image's lifetime of being shared over the internet and perhaps find a threshold between image quality, and the raw size of the image in bytes, using available image quality metrics. We will assess image quality using MSE, PSNR, SSIM, and other quality measurements used by leaders in the field of image quality metrics. We will be able to look at the difference between each image in Python and track the image quality as each generation is produced. This information could be used to make JPEG quality software for mass file resizing.

Introduction:

In the current age of high-speed digital communication and growing social media applications like Instagram and Twitter, the demand for optimal file size is needed to greatly increase the speed of data retrieval over the web. For text-based data, this process is relatively fast. However, for images, because of the larger size of the data, this procedure is slower. For example, A website's load-time is greatly affected by the size of the image and video files. Generally, the image file size will need to be reduced to new constraints, like reducing it in file size or rescaling it to fit the template of a website. This algorithm fundamentally changes the image from its source and affects the image pixel values.

The process of compressing image files from the non-source file occurs online every day with the sharing of information. An image can experience multiple generations of compressing. For this project we suggest that this will greatly degrade the subjective and objective image quality between the original source and therefore each generation of the new image.

Image Quality Assessment Overview:

The study of Image Quality Assessment is a growing field of research over the last few decades and is implemented in a wide range of applications. “Image quality measurement is crucial for most image processing applications”. (Wang, 2002) For the most part, there are three types of applications that use image quality metrics. The first is image quality handling software. An example would be a network media server that repeatedly monitors the quality of the digital transmission of a video stream. The second Is for testing image processing algorithms and used for optimization of the algorithm. Finally, the third is integrated systems for image quality assessment, like in digital cameras.

There are two ways to analyze image quality, subjective analysis, and objective analysis. Subjective analysis is based on a person’s opinion or perception. Because subjective analysis relies on human visual perception, forming an accurate quality metric is difficult and varies by a person’s own sensitivity. Subjective analysis requires multiple observers to judge and score image quality. For subjective analysis of image quality, this is called the “Mean Opinion Score (MOS) and has been used for many years” (Wang, 2002). This is not to say that subjective analysis is bad. In most cases, the human eye is the best judge of quality because ultimately the goal of an image quality algorithm is to try and simulate the human visual system (HVS).

Subjective analysis requires a large set of human observers to build accurate data. This process is costly and cannot be done in real-time. Because of these factors, objective analysis is largely used today in enterprise applications.

The process of objective quality assessment tries to measure the perceived image quality based on the human visual system and assess image quality with mathematical algorithms. Many algorithms have been developed over the years to measure image quality. These algorithms can fit into one of three groups; The first is a Full-Reference based calculation, which means a source image is used as the gold standard against a target image. The second is a No-Reference, and it only needs one image to judge quality, this is usually done with training data designed to look for types of image distortion. A Reduced-Reference analysis looks at a partial image to compute quality. For this project we will only be focused on Full-Reference based metrics.

Quality Assessment Metric:

The quality measurement algorithms we will be using for this experiment are as followed:

- **Mean Squared Error** (MSE): "The MSE value denotes the average difference of the pixels all over the image. A higher value of MSE designates a greater difference amid the original image and processed image." (Nadipally, 2019)
- **Peak Signal-to-Noise Ratio** (PSNR): "The PSNR calculates the PSNR ratio in decibels amid two images. We often use this ratio as a measurement of quality between the original image and the resultant image. The higher the value of PSNR, the better will be the quality of the output image." (Nadipally, 2019)
- **Universal Quality Image Index** (UQI): "The UQI evaluates quality of an image using loss of correlation, luminance distortion, and contrast distortion."(Oszust, 2016)
- **Structural Similarity Index** (SSIM): The SSIM is a perceptual metric that computes image quality loss, which can be caused by processing such as data compression.

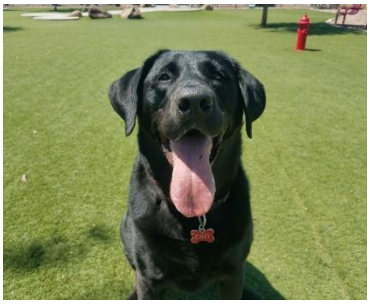
JPEG Compression Quality Overview:

JPEG image files use a changeable quality control with its lossy compression algorithm. Images saved at a lower quality have a smaller file size. Images saved at a high-quality setting will be larger in size. The range of quality settings in JPEG are 100 to 1. This range is usually depicted as a percentage in most documentation on JPEG. This perception of quality percentage is a bit misleading and will be addressed at the conclusion of the experiment.

Experiment:

During the age of analog tape media, such as VHS video and audio cassette tapes, quality degraded quickly between each generational copy. Currently, with digital data, this issue has been mitigated since you can make perfect digital copies with formats like PNG and BMP. However, with a lossy compression algorithm such as JPEG, used for saving images, one would expect that there will be some loss of data for each generational copy of an original source image. We have assumed from documentation that JPEG's Quality setting is based on a percentage. For the project we are testing JPEG lossy compression based on this VHS analogy to see the quality difference between generation to generation of copied image.

This project will examine the subjective, and objective quality loss of a compressed image. An image will be compressed and saved at a constant quality as a new image. Then this process is repeated on each new image over 50 generations. We will observe the objective difference in image quality metrics between each generation of compressed JPEG. We will measure each generation's image quality using image quality assessment algorithms and show the difference. We will chart the output with matplotlib in Python and compare quality to size on disk. We are looking to track image degradation over multiple generations at different quality levels. Then do a subjective observation of image quality on the data collected. It is our prediction that image quality degradation will compound with each generation and that our output will show this assumption.



The JPEG image used for this experiment has a size on disk of 1.36 MB (1,433,600 bytes).

Experiment Steps:

To speed up the time it took to run the full experiment, we split the python program into 5 smaller programs. This would make testing and fixing the code easier by setting up each step of the experiment. The steps are as followed:

1. The first program is called make_images.py. This program makes a copy of a reference image from the input directory. The program then compresses the copied image with a provided constant value. We call this constant value in the code the COMPRESS_SPECS. Then we save the image to the output directory. The output folder will get 50 generations of images at the constant quality setting. Each new generation is spawned from the last generation and each generation of JPEG is saved to the output file. For example, if the COMPRESS_SPECS is 75 then 50 generations of images will be saved in the output file at a quality level of 75, for a total of 50 images. The Python library OpenCV was used to set JPEG compression quality.

2. The second program used is called make_diff.py. Its role is to make a Difference image between the original image and each generation of compression quality JPEG save in the output folder. The make_diff.py output will be used in the presentation to highlight the visual difference between images. We also use the OpenCV library again in this program to subtract the difference between the two images.

3. The third program is called make_metrics.py. This program runs all 4-quality metrics discussed in our Quality Assessment Metric section and measures the size in bytes, of each generation of JPEG in the output file. The data is saved to a Python pickle file for the next program to chart.

This program uses the Python SEWAR library for measuring Full Reference images. Because of the sheer number of images measured, this process takes a long time to run. The algorithm for SSIM takes around 20 seconds alone to measure each image. Measuring all 300 images for this experiment took about an hour and a half to collect the data in this section.

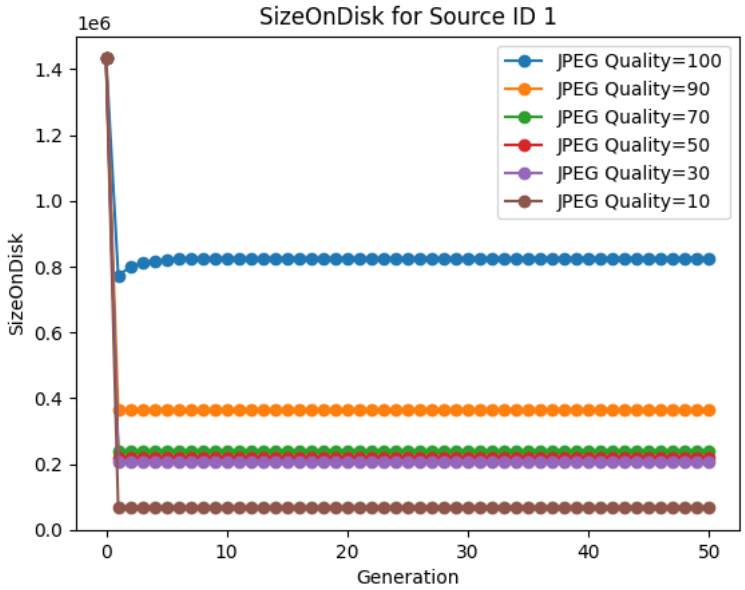
4. The fourth program is called make_charts.py. This program uses the pickle file stored in the metrics folder to create matplotlib charts to visualize our results data and show in the final presentation.

5. The last program is called info.py and was used to hold the quality setting values.

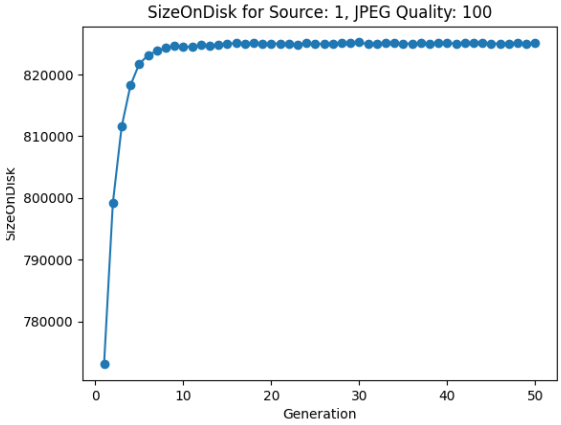
Experiment Results:

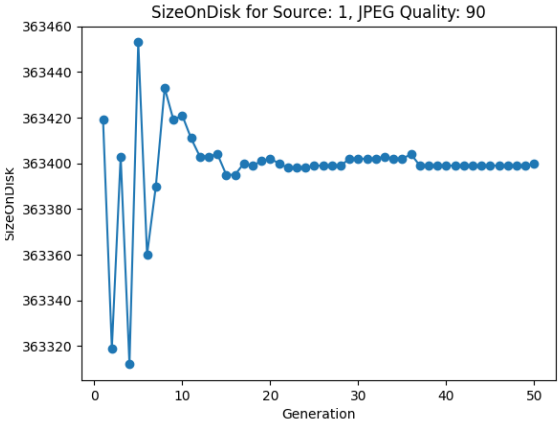
After collecting all the charted data, we found some interesting and unexpected results. It was our original assumption, that after compressing a JPEG over 50 generations, with a quality level set to less than 100, we would see a steady decline in measured quality and size on disk of the file. However, this is not what we observed after running six different quality values of compression, (100, 90, 70, 50, 30, 10) with 50 generations of compression.

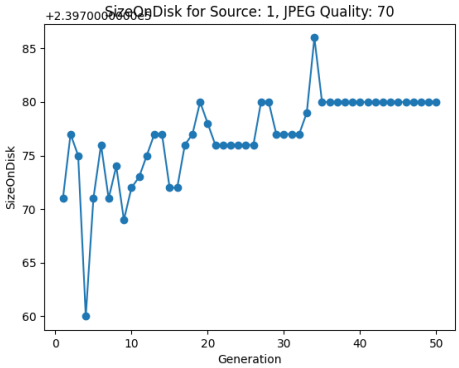
When we look at the chart for size on disk, the first thing we notice is the 100-quality setting in blue is almost half the memory size of the original source image. We did not expect to see any significant change in file size for the 100-quality set of generational data. Furthermore, we observe the size on disk does not scale to the JPEG's Quality setting. In our experiment we assumed JPEG quality was based on a percentage. However, it appears when JPEG's quality setting is adjusted, the JPEG compression algorithm will try and optimize the file size of the image by doing its own quality assessment based on the value given for compression.

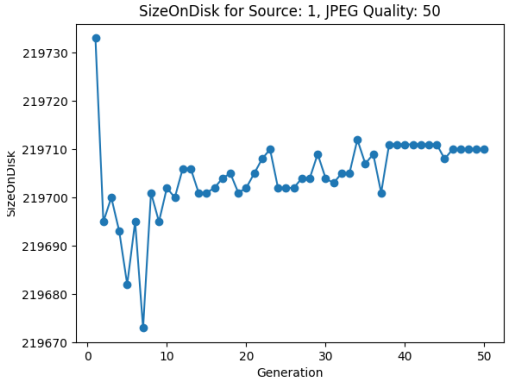


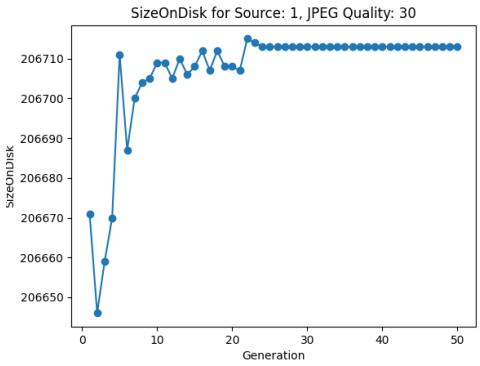
We observed a flatline of file size toward the end of generational compression on all JPEG Quality settings. This is better highlighted when we zoom in to each image size on disk. (Below size on disk)

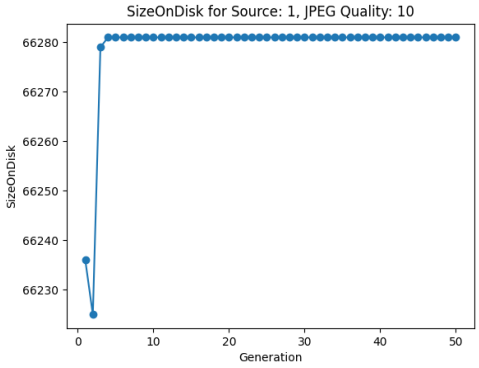






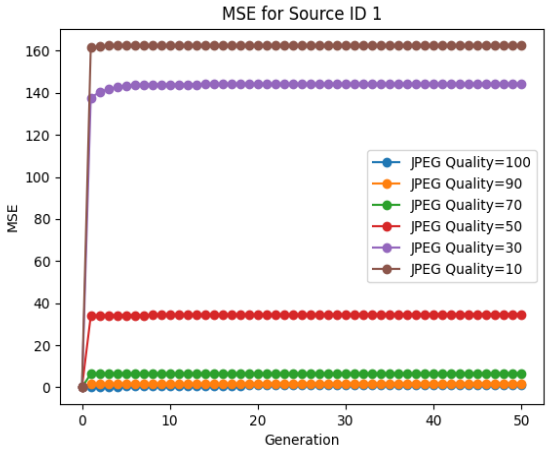




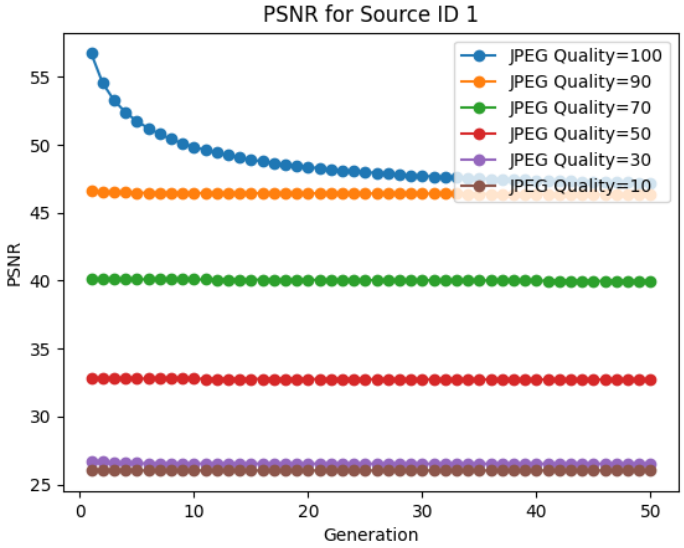


This data was completely unexpected. We expected to see the size on disk to slope downward with each generation of compression, for any quality setting not set to 100. If the JPEG quality setting was based on a percentage, then one would think each compressed copy would get smaller and smaller by compounding the quality percentage each generation.

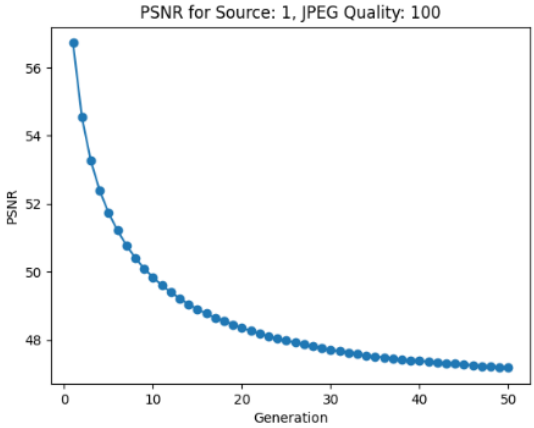
Image Quality Data:

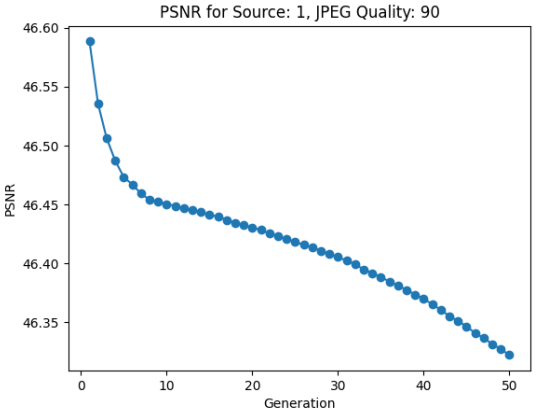


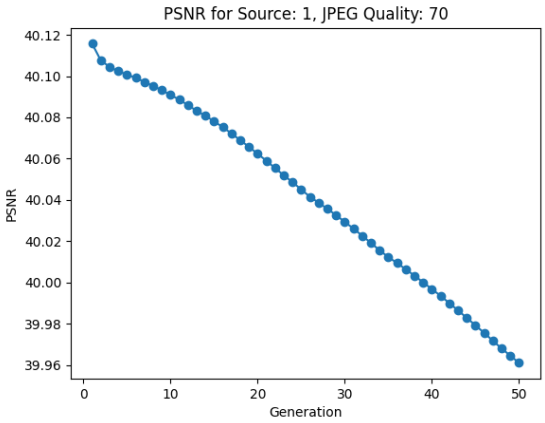
The quality metric data we collected for the experiment was a bit more predictable. The MSE value was higher on compressed images with a lower JPEG quality value. However, it is interesting to see the MSE gap between JPEG quality of 50 and 30. If we do subjective evaluation between the 50th generation at quality 50 and the 50th generation at quality 30, we see a difference in pixilation and colors. The 50th generation at quality 30 is much grainier in the image compared to the 50th generation at quality 50.

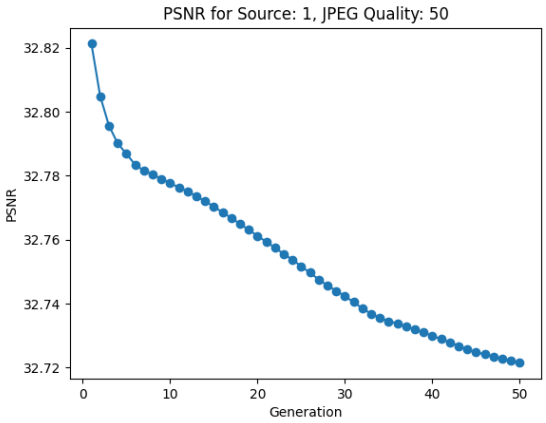


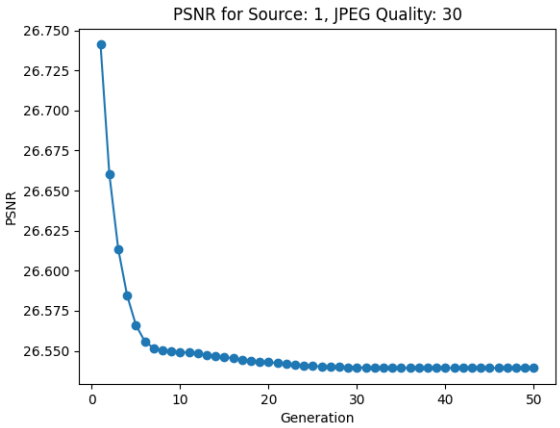
For PSNR , we don't see anything unexpected in the data at the view above. However, when we zoom into each quality setting, we do find that PSNR is sloping downward on all quality levels. This shows that PSNR is working and can measure the small changes caused after each compressed generation.

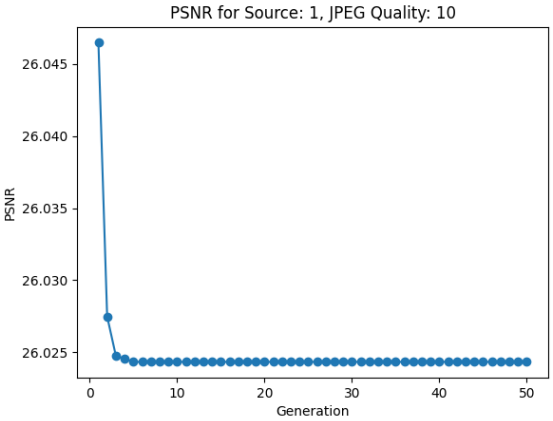






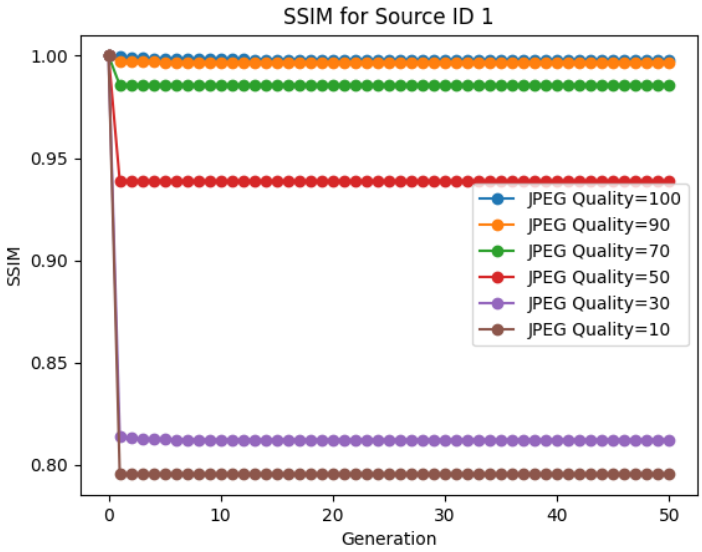




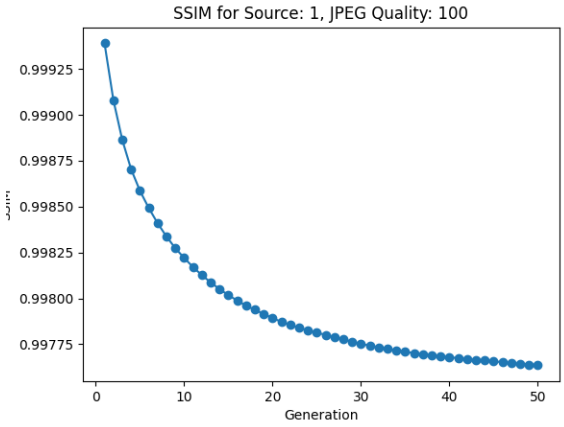


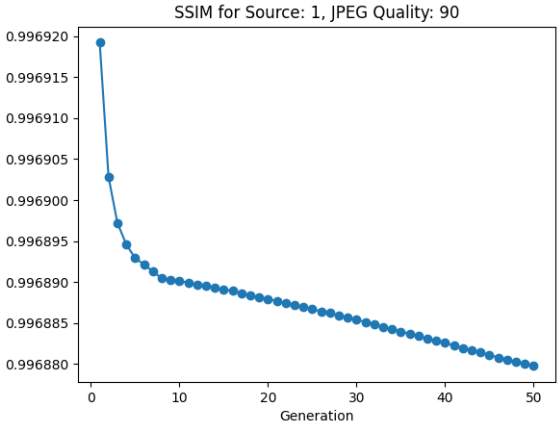
The charts for SSIM and UQI are very similar looking. But I found that SSIM gives us better information in terms of trying to define a threshold value of difference the HVS can translate subjectively.

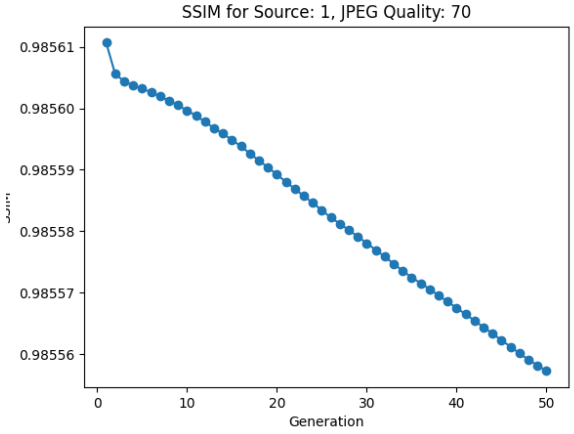
The image being compressed with the quality level of 70 or greater are subjectively identical to each other and the original source image.

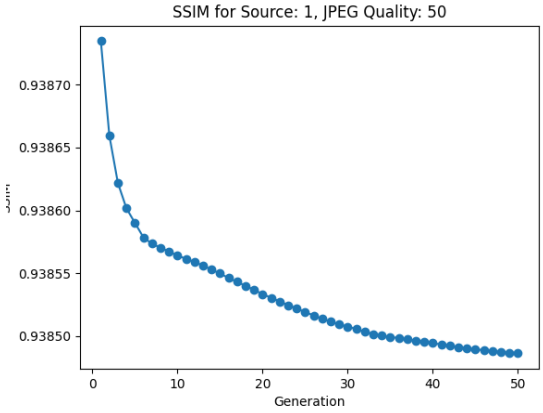


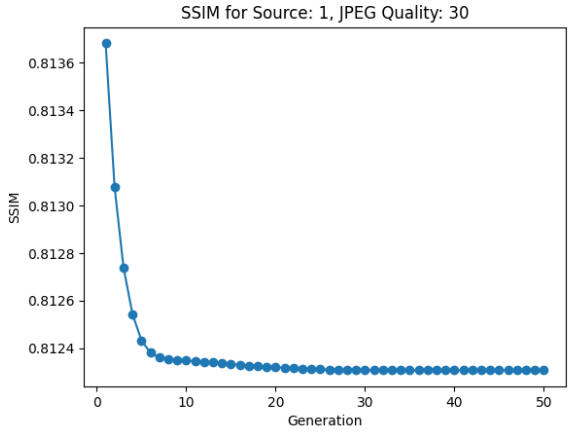
When we Zoom out on SSIM we are met with the same downward trending quality we saw in the PSNR graphs. In fact, the charts are almost identical to PSNR.

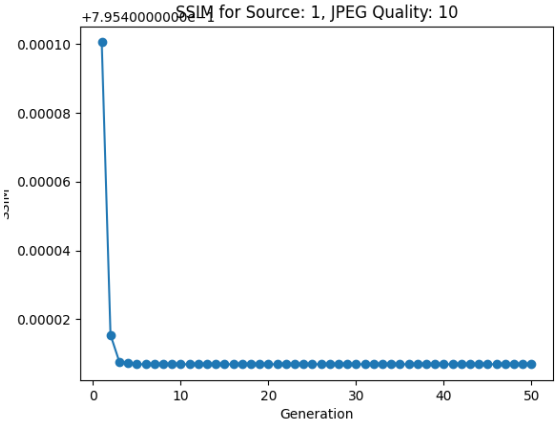




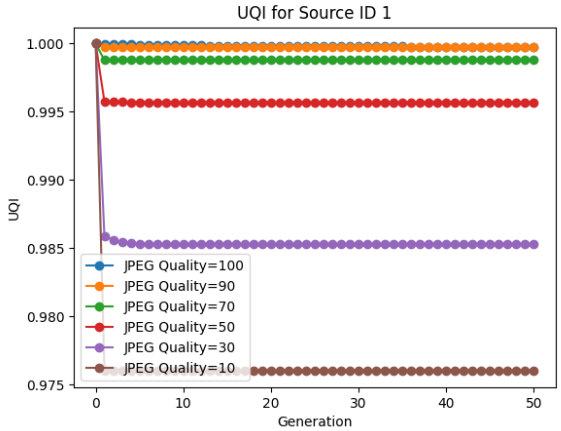








As stated above UQI gives us a similar metric as SSIM. However, we found UQI clumps together images set at a 50-quality compression to the upper range of the UQI metric. Subjectively the differences in quality at 50 were noticeably perceived with HVS. Therefore, we thought SSIM was a better tool for basing a size on disk comparison to quality.



Conclusion:

When designing this experiment some key assumptions were made about the quality level settings of JPEG compression. The first assumption was that the quality setting was a percentage of change to the current file size. The second assumption was that JPEG image quality set at 100 would not affect the size on disk. JPEG will optimize the image file to save space. The JPEG format will optimize size over quality. This is perhaps why we saw a flat-line effect on the size on disk charts. Once JPEG has optimized the file over and over so many times it can't save anymore space based on the requested quality. It appears this means when an image is compressed at a level of 100, JPEG will try and optimize the space of the new file based on their own image quality meter. This may be the reason the file size was almost cut in half after a 100-quality compression. It also appears JPEG tries to clean up the image after compression. We found that images at the 70-quality look subjectively less blurry around the edges of structures compared to the original source image. For example, around the edge of the dog or the fire-hydrant in the background of the image.

Evaluation of Approach:

In this project there were so many unforeseen problems when it came to the difficulty of Image Quality Assessment. This is a hard field to understand. This group did not have the best mathematical background to fully understand the complexities of the IQA algorithms like SSIM. Therefore, we used a library called SEWAR for these image quality metric functions. We also found we should have done more research on the JPEG format. We may have been better off not going down this road at the start of the project and just showing images that had their quality affected by distortion. The goal was to find a threshold value of quality to help with file size reduction and I do feel we have good evidence to show SSIM and PSNR are good metrics for this calculation. However, this group was not able to create an application to do this as envisioned. I believe this had to do with only having two members in the group at the start of the assignment. Along with doing so much legwork at the beginning of the assignment that was centered around coding Image Quality Assessment tools from scratch.

References:

Z. Wang, A. C. Bovik and L. Lu, "Why is image quality assessment so difficult?," 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002, pp. IV-3313-IV-3316, doi: 10.1109/ICASSP.2002.5745362.

Nadipally, M. (2019). Intelligent Data Analysis for Biomedical Applications. Academic Press.

Oszust M. (2016). Full-Reference Image Quality Assessment with Linear Combination of Genetically Selected Quality Measures. PloS one, 11(6), e0158333. https://doi.org/10.1371/journal.pone.0158333