Data quality assessment - Cost

Victor Stenmark

July 23, 2025

1 Introduction

For this calculation, I used a selection of the available NR-IQA methods from `pyiqa`. I chose to exclude methods of the same form but with different training data or implementation. So I only included `brisque` and not `brisque_matlab`. I also excluded methods that are not general NR-IQA methods and `ahiq`, as it did not work for some reason, and `pi`, as it was incredibly slow. This leaves me with these methods:

```
['arniqa', 'brisque', 'clipiqa', 'cnniqa', 'dbcnn',
'hyperiqa', 'ilniqe', 'liqe', 'maniqa', 'musiq', 'nima', 'niqe',
'nrqm', 'paq2piq', 'pi', 'piqe', 'qalign', 'qualiclip',
'topiq_nr', 'tres', 'tres-flive', 'unique', 'wadiqam_nr', 'qalign']
```

All of these methods except for Q-Align could run on the RISE machine, which has two GeForce RTX 2070 SUPER with 8GB of RAM. I ran the NR-IQA methods on 1,000 images from the dataset I used in my thesis. I used a batch size of 8 images and loaded them into memory as needed. I could probably have used a larger batch size, and it would have been faster to load the images once and then store them in a large tensor. However, some models may not support batch sizes larger than eight, and loading all the images at once is not possible for massive image sets (one could implement a nice caching strategy if speed is a concern). Given these things, my timing estimates are likely upper limits.

2 Q-Align

As previously mentioned, I was unable to get Q-Align working on the RISE machine due to out-of-memory errors on the GPU. This was despite clearing cache memory, using `torch.no_grad`, and a batch size of 1.

I instead ran Q-Align on Google Colab using an L4 GPU with 22.5 gigabytes of GPU RAM. I used a batch size of 1 and, like the previous run, loaded the images as needed. This machine costs roughly 2.09 units per hour to run, and each 100 credits in Google Colab costs £11.56.

3 Results

Figures 1 and 2 show the inference and total wall times for running the various metrics. The inference times represent the accumulated time the model is spending generating the score. The wall time refers to the total time it took to load and score the 1000 images.

Starting with the cost to assess the images, the only cost stems from the Colab credits required to get the specialized hardware to run Q-Align. Given that it consumes 2.09 credits per hour and takes 0.3575 (1287/3600) hours to run 1000 images, the total number of credits consumed is $2.09 \times 0.3575 = 0.747$. Thus, it costs 0.747 credits per 1000 images, meaning that scoring one image costs 0.000747 credits. With 100 credits, it's possible to get $\frac{100}{0.000747} \approx 133868$ images for only £ 11.56. This is also an upper limit, as increasing the number of images would likely be much cheaper, as indicated by the significant time difference between the total wall time and the inference time of Q-Align. Loading the model takes some time, but once that is finished, scoring the images goes fairly quickly.

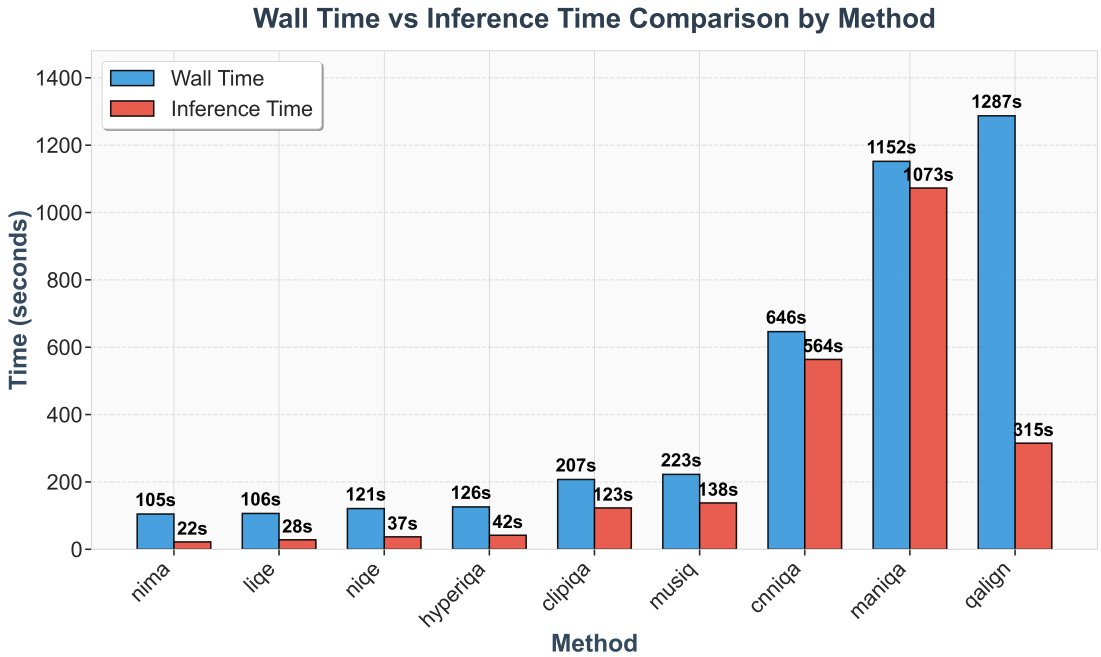

Figure 1: Comparison of wall time and inference time for 10 NR-IQA methods

Moving on to the time estimates, I must reiterate that the following numbers will likely be overestimates, as I didn't use a large batch size and did not cache images. The total time of the test was 2 hours and 23 minutes, which is a lot for 1000 images, but it is not necessary to use all methods. If one were to use

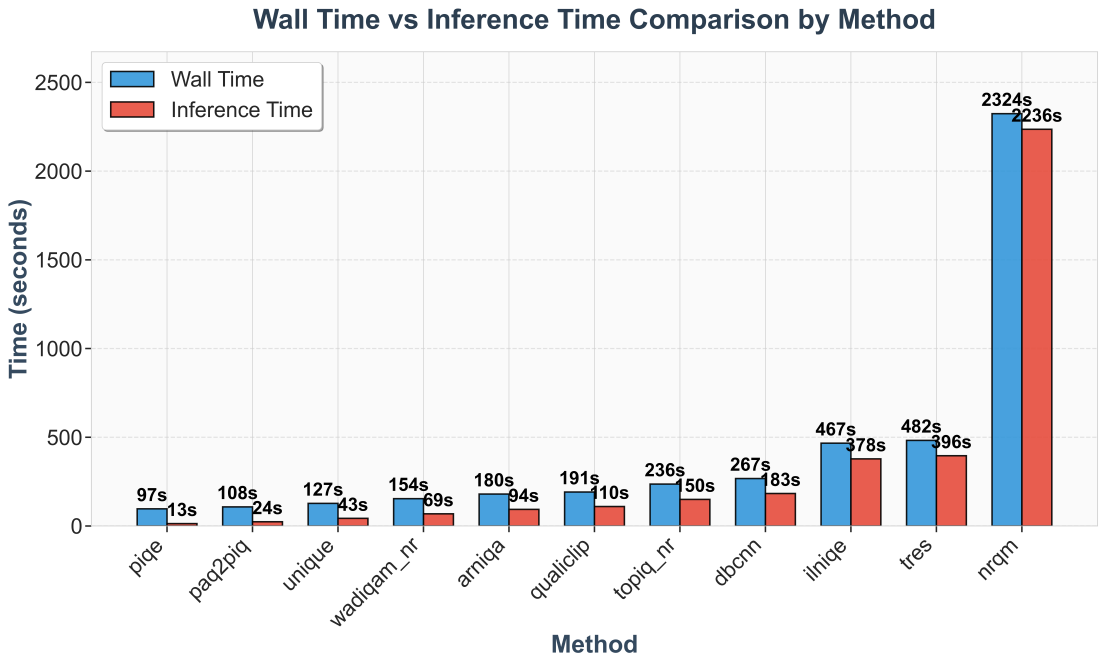

Figure 2: Comparison of wall time and inference time for 10 NR-IQA methods

only a subset of the methods, say: `ilniqe, qalign, topiq_nr, musiq`, then it would take 36 minutes or 2.2 seconds per image. One could even exclude Q-Align and then reduce the time to 0.9 seconds per image.

4 Conclusion

The results show that scoring the images is both very cost-effective and can also be completed in a reasonable amount of time. One thing to consider is that using Q-Align requires some additional configuration, as Google Colab has to be able to access the images. It is probably easiest to do so via uploading the images to Google Drive, as Colab integrates directly with Google Drive, but this might be tricky for huge image sets. Still, using Q-Align is, in my opinion, worthwhile, as it has proven to be very effective at NR-IQA.