A Python Framework for Objective Visual Quality Assessment

Caio L. Saigg, Bruno S.S. Dias,
André H. M. Costa, and Mylène C.Q. Farias
Department of Electrical Engineering
University of Brasília
Brasília, Brazil

Helard B. Martinez
School of Computer Science
University College Dublin
Dublin, Ireland

***Abstract*—This work introduces a Quality Assessment Framework that provides researchers with the flexibility, consistency, and scalability they need to evaluate and compare quality metrics, promoting the reproducibility of results. The framework is open source (Python) and currently has 11 visual quality metrics that use 3 different libraries: Scikit-video, FFmpeg toolkit, and PyMetrikz. It can be easily expanded to include more metrics in the future and allows testing on several quality datasets. To validate it, we tested it on two datasets and compared the results with the results obtained by other authors in the literature. The results are consistent with those reported by external studies. With this evidence, new image/video metrics and datasets can be integrated into this framework. This will allow researchers to compare their methods with a wide number of quality metrics on several datasets in a fast and efficient way.**

I. Introduction

In recent decades, there has been a tremendous increase in the popularity of multimedia applications, with 82% of Internet traffic currently being video data [1]. At the same time, these applications have been incorporated into human daily activities. For example, video streaming and online gaming are widely used for entertainment, while video conferencing has become essential for remote work, especially during the COVID-19 pandemic. Therefore, ensuring high levels of trustworthiness and Quality of Experience (QoE) is now critical to guarantee the success of these applications. Since the success or popularity of a service is correlated with the QoE of the user, it is important to have tools to estimate the quality of the signal on the client side [2].

Subjective experiments, in which human participants rate the perceived quality of a set of test videos, are considered the most accurate way to measure QoE. However, they are expensive and time consuming and, therefore, hard to incorporate to real-time applications. A better alternative is the use of objective quality metrics, which are computational algorithms that estimate signal quality. Depending on the amount of reference information (source) used, objective quality metrics can be classified as Full-Reference (FR), Reduced-Reference (RR), and No-Reference (NR) methods. FR methods require the reference and processed signal to estimate quality, RR methods require a small amount of the reference (e.g. attribute measures) and the processed content, and NR methods require only the processed content. Currently, several objective quality assessment methods have been proposed in the literature for audio, images, and videos [3]. Therefore, it is often difficult to compare their performance for different datasets and applications.

This paper presents a quality assessment framework to evaluate and compare different visual quality metrics (image and video). The framework seeks to create a common and flexible assessment pipeline for image and video quality metrics that allows one to compare metrics fairly while promoting the reproducibility of the results.

sectionRelated Works

The use of frameworks is an advancement in the development of quality metrics. It allows faster training and testing of metrics, as well as easier comparison of the results. Because image and video quality assessment requires working with large files, training and testing quality metrics can take a long time. Therefore, a reduction in this time can represent an important gain in productivity. In addition, a framework that contains several quality metrics can facilitate the comparison with the state-of-the-art. It is worth noting that software frameworks have been developed in different areas of signal processing. For example, in audio signal processing, Geraghty *et al.* [4] implemented a platform for objective speech and audio quality metrics.

In the evaluation of visual image and video quality, Murthy and Karam [5] proposed a MATLAB-based framework. Although MATLAB has been widely used in many engineering fields, its use has been outgrown by other programming languages such as Python. By utilizing an open-source programming language, our work facilitates the access and reproducibility of different metrics. The Python languages possess a large number of scientific libraries such as Numpy and Matplotlib. These libraries facilitate the implementation of quality metrics. Libraries such as Scikit-learn, Tensorflow, and Pytorch present many tools for the implementation of machine learning-based metrics. Garcia *et al.* [6] propose a framework for the quality assessment of video and audio content in WebRTC applications. Their work focuses on the comparison of FR metrics, simulating many network conditions. On the other hand, our framework contains both FR and NR video

978-1-6654-5385-1/22/$31.00 ©2022 IEEE

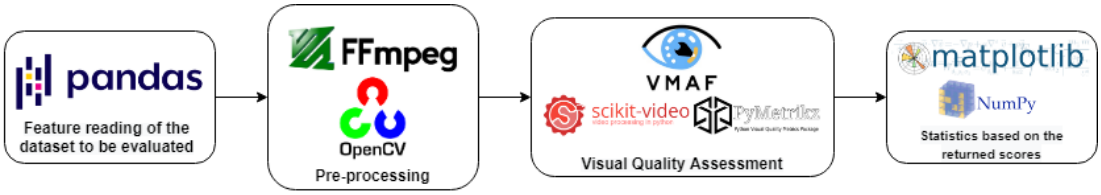


Fig. 1: Block diagram of the Quality Assessment Framework.

TABLE I: Sample lines of a *csv* file used to run the framework.

refFile	testFile	MOS	height	width
bus	bus_dirac_1	0.914	1088	1920
bus	bus_dirac_2	1.761	1088	1920
bus	bus_dirac_3	2.545	1088	1920
bus	bus_h264_1	-0.06	1088	1920
bus	bus_h264_2	0.293	1088	1920
bus	bus_h264_3	0.425	1088	1920
bus	bus_h264_4	1.709	1088	1920
bus	bus_mpeg2_1	0.037	1088	1920
bus	bus_mpeg2_2	0.614	1088	1920

quality metrics. A broader selection of metrics allows for a richer benchmarking of video QoE metrics.

II. Framework Description

The purpose of this work was to develop a video quality estimation framework that is easily scalable. With this goal, the whole structure of the framework was assembled such that it is possible to easily add new audiovisual objective quality metrics, which makes the framework a conglomerate of several quality metrics. This framework makes it easier to test and compare objective quality metrics on several quality datasets. The framework was developed using Python 3 programming language. The software was developed on a Linux operating system, using a container platform. The framework is easy to install on shared computers or servers that can be easily accessed by several users. The framework code is currently available, but it is worth highlighting that the documentation, as well as the organization of the codes and files, are not yet finalized. In addition to Python libraries, such as Pandas and Numpy, the framework also uses the FFmpeg toolkit.

Figure 1 depicts the various stages of the implemented framework. To run the software for several quality metrics and datasets, we first need to organize the quality datasets, separating the source and processed video sequences into two different folders. After separating the contents, we should create a *csv* file in the same directory of the code, containing the filenames of all the source and processed video sequences. More specifically, each line of the *csv* file should contain the filename of the source content, the filename of the processed content, the mean observer score (MOS) provided by the quality dataset, the height and width (spatial resolution) of the video sequences, as shown in the example of the first lines of a typical *cvs* file shown in Table I.

Next, the user should build a *json* file that contains the information necessary to run the framework. More specifically, the file should contain the filename of the *csv* file, the format of the videos in the quality dataset, the quality metrics to be used, and the paths to the folders with the source and processed video sequences. When running the framework, it is also possible for the user to edit the original *json* file by adding "-edit" to the command line. Currently, 11 visual quality metrics are included in the framework, but more metrics can be easily added to the framework. These metrics were obtained from three different sources: 5 are from the scikit-video library, one (VMAF) from the FFmpeg toolkit, and 5 from the PyMetrikz library.

The quality metrics taken from the Scikit library are: the Structural Similarity Index (SSIM) [7], the Multiscale SSIM (MS-SSIM) [8], the Peak Signal-to-Noise Ratio (PSNR), the Mean Squared Error (MSE), and the Naturalness Image Quality Evaluator (NIQE) [9]. With the exception of NIQE, which is an NR image quality metric, all other metrics are FR quality metrics. For FR quality metrics, the software uses both source and processed contents, comparing them. However, for NR quality metrics, the software will only use the processed content to compute the predicted score.

The Video Multi-Method Assessment Fusion (VMAF) [10], [11] is an objective video quality metric developed by Netflix. The metric targets videos processed with different video codecs, different encoding configurations, or transmission protocols. To include VMAF in the proposed framework, it was necessary to use the FFmpeg toolkit and a Python script to call the VMAF code.

The last set of quality metrics was taken from the PyMetrikz library, which is a Python package that implements various quality metrics. The metrics included in the framework are: Root Mean Squared Error (RMSE), Signal-to-noise ratio (SNR), Weighted SNR (WSNR), the Universal Quality Index (UQI) [12], and the Visual Information Fidelity (VIF) [13]. These are all FR quality metrics. Since PyMetrikz was designed for images, to make it work with videos, we implemented the modifications depicted in Figure 2. The modifications consisted of first checking if the video format is *yuv*. If so, a conversion from *yuv* to *avi* format is performed using FFmpeg. Next, we use the OpenCV library to separate video frames into images in PNG format. Finally, we run PyMetrikz for each of the frame pairs (source and processed), save these

https://github.com/Scholles007/Framework-for-Objective-Visual-Quality-Assessment-FOVQA

https://gitlab.com/gpds-unb/pymetrikz

Fig. 2: Block diagram for using the PyMetrikz-base image quality metrics to estimate the quality of videos.

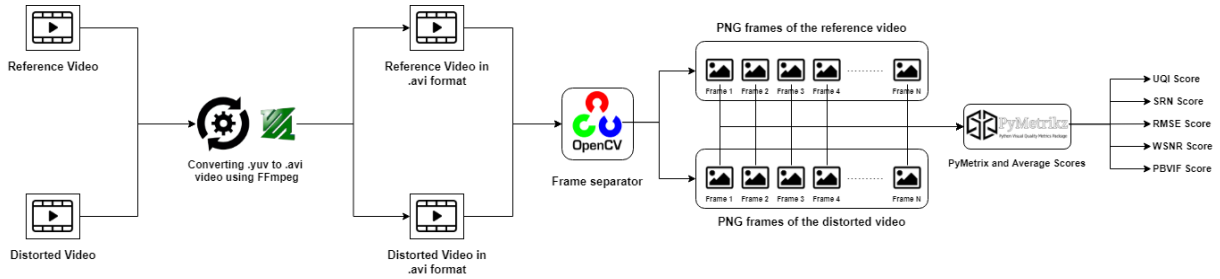


TABLE II: Table showing the output *csv* file for a set of quality metrics for a set of processed video sequences.

refFile	testFile	MOS	height	width	ssim	msssim	psnr	mse	vmaf	rmse	snr	wsnr	uqi	pbvif	niqe
bus	bus_dirac_1	0.914	1088	1920	0.98	0.98	35.43	19.12	96.57	0.025	26.79	36.87	0.54	0.94	12.06
bus	bus_dirac_2	1.761	1088	1920	0.97	0.97	33.36	30.63	84.86	0.031	24.90	32.68	0.46	0.87	12.57
bus	bus_dirac_3	2.545	1088	1920	0.94	0.94	30.03	65.69	63.76	0.042	22.19	27.58	0.37	0.72	13.20
bus	bus_h264_1	-0.06	1088	1920	0.99	0.99	37.06	12.97	99.49	0.019	29.00	42.20	0.62	1.00	11.34
bus	bus_h264_2	0.293	1088	1920	0.98	0.99	36.40	15.09	96.26	0.021	28.20	38.27	0.56	0.97	12.10
bus	bus_h264_3	0.425	1088	1920	0.98	0.98	35.74	17.67	91.55	0.023	27.48	35.99	0.52	0.95	12.20
bus	bus_h264_4	1.709	1088	1920	0.96	0.97	33.94	27.05	79.55	0.028	25.78	31.97	0.45	0.90	12.43
bus	bus_mpeg2_1	0.037	1088	1920	0.99	0.99	37.03	13.12	99.90	0.020	28.75	43.41	0.61	1.00	11.29
bus	bus_mpeg2_2	0.614	1088	1920	0.98	0.98	34.48	23.70	87.80	0.026	26.36	35.67	0.47	0.92	13.64
bus	bus_mpeg2_3	1.5951	1088	1920	0.97	0.98	32.79	35.07	78.49	0.031	24.90	32.66	0.42	0.88	13.59

TABLE III: Sample statistics output for IVPL [14] Dataset.

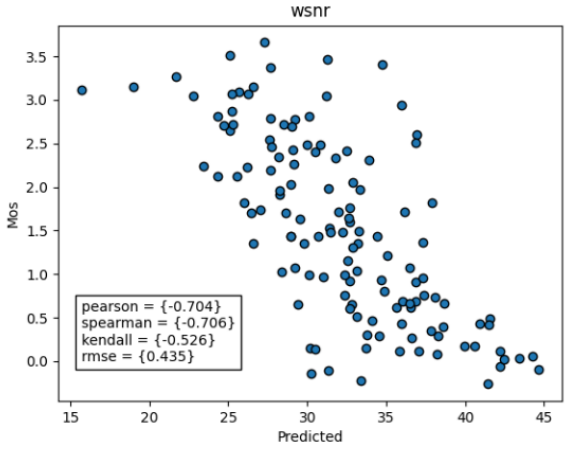
	Pearson	Spearman	Kendall	RMSE
ssim	0.573	0.642	0.471	0.549
msssim	0.589	0.734	0.554	0.569
psnr	0.647	0.658	0.485	0.437
niqe	0.127	0.157	0.118	0.307
vmaf	0.608	0.611	0.462	0.518
rmse	0.609	0.644	0.491	0.289
snr	0.389	0.390	0.297	0.414
wsnr	0.704	0.706	0.526	0.435
uqi	0.124	0.211	0.176	0.388
pbvif	0.465	0.431	0.324	0.473

scores into an array, and average them to obtain a quality score for the video.

The framework outputs the results of the metrics in the same *csv* file informed in the *json* file. In the flow diagram in Figure 1, there is a step to insert new columns into the *csv* file. For each quality metric, a new column is added with the output quality scores. But a new column is only added if it is not already in the file. In the same way, only empty spaces in the table are filled. In other words, the framework does not recalculate the output values of processed videos. In this way, if the execution of the program is interrupted before it finishes the complete batch of metrics and video sequences, the user can restart it and the program will resume where it stopped. Table II shows some rows of a sample output file, where columns 6-16 correspond to the quality metrics tested.

After the framework has finished computing all quality scores, it performs a statistical analysis, considering the subjective quality scores provided with the quality datasets. The following statistics are computed: Pearson's linear correlation coefficient (PLCC), Spearman rank order correlation coefficient (SROCC), Kendall rank order correlation coefficient

Fig. 3: Example scatter plot for the WSNR metric for the IVPL Dataset.



(KROCC), and the Root Mean Squared Error (RMSE) [15], [16]. The outputs of the PLCC, SROCC, and KLCC metrics range from 0 to 1, with values closer to 1 representing better performances of the quality metrics. For the RMSE lower values represent better performances. These statistics are saved in a new *csv* file which contains the results of all the tests performed. Scatter plots are also saved to illustrate these results. Table III shows the output of the generated sample statistics, while Figure 3 shows a sample scatter plot.

III. Validation Results

To verify whether the implemented framework delivers reliable outputs, we compared the PLCC and SROCC values obtained for two datasets with the results published in the

TABLE IV: Comparison between the statistics of the framework and of other works in the literature.

Database	Metrics	Helmrich et al. [17]		Lin et al. [18]		Wu et al. [19]		Liu et al. [20]		Framework	
		PLCC	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC	PLCC	SROCC
LIVE	VMAF	0.729	0.752			0.761	0.754			0.729	0.752
	SSIM	0.626	0.694	0.607	0.567			0.710	0.696	0.628	0.696
	MSSSIM	0.675	0.732	0.685	0.677	0.743	0.736	0.743	0.736	0.692	0.750
	PSNR	0.539	0.523	0.573	0.414			0.539	0.523	0.537	0.518
	NIQE			0.331	0.176			0.196	0.019	0.027	0.079
	WSNR									0.667	0.635
	UQI									0.391	0.410
	PBVIF									0.383	0.409
IVPL	VMAF	0.591	0.580			0.591	0.579			0.608	0.611
	SSIM	0.570	0.635	0.819	0.804			0.689	0.685	0.573	0.642
	MSSSIM	0.546	0.574	0.828	0.791	0.595	0.579	0.710	0.706	0.589	0.733
	PSNR	0.632	0.647	0.799	0.815			0.707	0.713	0.647	0.657
	NIQE			0.395	0.235			0.301	0.245	0.127	0.157
	WSNR									0.704	0.706
	UQI									0.124	0.211
	PBVIF									0.465	0.431

literature. The video quality datasets were the LIVE [20] and IVPL [14] datasets. Table IV shows this comparison. Notice that the results obtained with the framework are very similar to those obtained by other authors [17]–[20]. The NIQE metric was the metric that presented the greatest divergence.

In Table II, the MOS column is completed using the DMOS (Difference MOS) values obtained from the IVPL dataset. DMOS is the difference between the reference and processed mean opinion scores in a full reference test. This means that the lower the DMOS value, the smaller the difference between the quality of the reference and processed video and, therefore, the higher the quality of the processed video (assuming the video with the maximum quality is the reference video). We can see that for all distortions, as we increase the reference number (1, 2, 3, and 4) that corresponds to decreasing values of bitrates, the DMOS value increases. This is illustrated in Figure 4, which shows an original video frame and the 3 corresponding frames taken from MPEG2-compressed videos with 3 decreasing bitrates (lines 10, 11, and 12 in Table II). Therefore, the tests in Table II show that, for all metrics, as we increase the distortion (or decrease the compression bitrate), as expected, the final quality scores tend to decrease. In summary, the framework fulfills its role of returning quality scores that are consistent with the quality of the videos.

IV. Conclusion

In this paper, we introduce a framework for visual quality assessment. The framework was developed to provide flexibility, repeatability, and scalability for researchers who need to test and compare objective quality metrics on several quality datasets. The framework was implemented in Python, which is an open-source programming language with a large number of libraries available. Currently, the framework has 11 visual quality metrics obtained from three different sources, the scikit-video library, the FFmpeg toolkit, and the PyMetrikz library. More metrics can be easily added to the framework. We validated the framework by testing it on two datasets and comparing the results (in terms of correlation coefficients) with the results published in the literature. Future work includes adding additional quality metrics and flexible input formats, such as 3D and 360-degree videos.

https://live.ece.utexas.edu/research/LIVE_APV_Study/apv_index.html
https://ivp.ee.cuhk.edu.hk/research/database.shtml

V. Acknowledgements

This work has been supported by the Brazilian National Council for Scientific and Technological Development (CNPq) and the University of Brasília (UnB)

References

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022," February 2019.

[2] H.-C. Soong and P.-Y. Lau, "Video quality assessment: A review of full-referenced, reduced-referenced and no-referenced methods," in *2017 IEEE 13th International Colloquium on Signal Processing & its Applications (CSPA)*. IEEE, 2017, pp. 232–237.

[3] Q. Fan, W. Luo, Y. Xia, G. Li, and D. He, "Metrics and methods of video quality assessment: a brief review," *Multimedia Tools and Applications*, vol. 78, no. 22, pp. 31 019–31 033, 2019.

[4] J. Geraghty, J. Li, A. Ragano, and A. Hines, "Aqp: An open modular python platform for objective speech and audio quality metrics," *arXiv preprint arXiv:2110.13589*, 2021.

[5] A. V. Murthy and L. J. Karam, "A matlab-based framework for image and video quality evaluation," in *2010 Second International Workshop on Quality of Multimedia Experience (QoMEX)*, 2010, pp. 242–247.

[6] B. García, F. Gortázar, M. Gallego, and A. Hines, "Assessment of qoe for video and audio in webrtc applications using full-reference models," *Electronics*, vol. 9, no. 3, p. 462, 2020.

[7] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[8] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402.

[9] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal processing letters*, vol. 20, no. 3, pp. 209–212, 2012.

[10] J. Y. Lin, T.-J. Liu, E. C.-H. Wu, and C.-C. J. Kuo, "A fusion-based video quality assessment (fvqa) index," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, 2014, pp. 1–5.

[11] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a practical perceptual video quality metric," 2016. [Online]. Available: https://medium.com/netflix-techblog/toward-a-practical-perceptualvideo-quality-metric-653f208b9652

(a) (b) (c) (d)

Fig. 4: Sample video frames with several quality levels: (a) original frame and (b,c,d) 3 corresponding frames taken from MPGE2-compressed videos with decreasing bitrates (lines 10, 11, and 12 in Table II).

[12] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Transactions on image processing*, vol. 15, no. 2, pp. 430–444, 2006.

[13] ——, "A visual information fidelity approach to video quality assessment," in *The first international workshop on video processing and quality metrics for consumer electronics*, vol. 7, no. 2. sn, 2005, pp. 2117–2128.

[14] F. Zhang, S. Li, L. Ma, Y. C. Wong, and K. N. Ngan, "Ivp subjective quality video database," *The Chinese University of Hong Kong, http://ivp. ee. cuhk. edu. hk/research/database/subjective*, 2011.

[15] H. J. K. James Algina, "Comparing squared multiple correlation coefficients: Examination of a confidence interval and a test significance." *Psychological Methods*, no. 1939-1463, pp. 76–83, 1999.

[16] P. Bobko, *Correlation and regression: Applications for industrial organizational psychology and management (2nd ed.).* CA: Sage Publications, 2001.

[17] C. R. Helmrich, M. Siekmann, S. Becker, S. Bosse, D. Marpe, and T. Wiegand, "Xpsnr: A low-complexity extension of the perceptually weighted peak signal-to-noise ratio for high-resolution video quality assessment," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2727–2731.

[18] L. Lin, J. Yang, Z. Wang, L. Zhou, W. Chen, and Y. Xu, "Compressed video quality index based on saliency-aware artifact detection," *Sensors*, vol. 21, no. 19, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/19/6429

[19] J. Wu, Y. Liu, W. Dong, G. Shi, and W. Lin, "Quality assessment for video with degradation along salient trajectories," *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2738–2749, 2019.

[20] Y. Liu, J. Wu, L. Li, W. Dong, J. Zhang, and G. Shi, "Spatiotemporal representation learning for blind video quality assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 3500–3513, 2021.