IQM-Vis: A User-Centric Python Toolbox for Visualising and Evaluating Image Quality Metrics

Matt Clifford[a*], Alexander Hepburn[b], Ricardo Kleinlein[c], Jorge Vila-Tomás[d], Pablo Hernández-Cámara[d], Paula Dauden[d], Nathan Lepora[b], Valero Laparra[d], Raúl Santos-Rodríguez[b]

[a]School of Computer Science, University of Bristol, Merchant Venturers Building, 75 Woodland Road, Bristol, BS8 1UB, United Kingdom
[b]School of Engineering Mathematics and Technology, University of Bristol, Ada Lovelace Building, Tankard's Close, Bristol, BS8 1TW, United Kingdom
[c]Department of Anesthesiology, Perioperative and Pain Medicine, Brigham and Women's Hospital, Harvard Medical School, 75 Francis St, Boston, MA 02115, United States
[d] Image Processing Lab, Universitat de Valencia, E4 building - 4th floor, Parc Científic Universitat de València, C/ Cat. Agustín Escardino Benlloch, 9 46980 Paterna (València). Spain
[*]Corresponding author email: matt.clifford@bristol.ac.uk

Abstract
Image Quality Metrics (IQMs) assess differences between images where human judgements are too expensive or infeasible due to the number of evaluations needed. Although a multitude of IQMs have been proposed in the literature, none are considered flawless. Therefore, it is crucial to understand their limitations by evaluating their suitability for different application domains. This type of evaluation is both qualitative and quantitative, lending itself best to interactive graphical tools. To this end, we created IQM-Vis, the first open source toolbox dedicated to analysing IQMs, visualising image distortions and conducting human image perception experiments, all through a simple Python interface.

Keywords
Image Quality Metrics; Human Image Perception; Perceptual Distances

Code metadata

Nr.	Code metadata description	Please fill in this column
C1	Current code version	v0.2.5.87
C2	Permanent link to code/repository used for this code version	https://github.com/mattclifford1/IQM-Vis
C3	Permanent link to Reproducible Capsule	N/A
C4	Legal Code License	BSD-3-Clause
C5	Code versioning system used	Git
C6	Software code languages, tools, and services used	Python
C7	Compilation requirements, operating environments & dependencies	Linux, macOS, or Windows; Python 3.9+; Python package dependancies: https://github.com/mattclifford1/IQM-Vis/blob/main/requirements.txt
C8	If available Link to developer documentation/manual	https://mattclifford1.github.io/IQM-Vis/
C9	Support email for questions	matt.clifford@bristol.ac.uk

1. Introduction

Image quality metrics (IQMs) serve as an objective evaluation of the perceived quality of an image. IQMs aim to capture how humans perceive differences between images. For example in Figure 1, reference IQMs compare a reference image and the same image after a distortion is applied. The goal is to recreate how humans perceive this difference according to human psychophysical experiments. They are utilised in scenarios such as regularisation of deep learning models [7] and benchmarking the performance of image processing algorithms where human evaluation is too expensive [8].

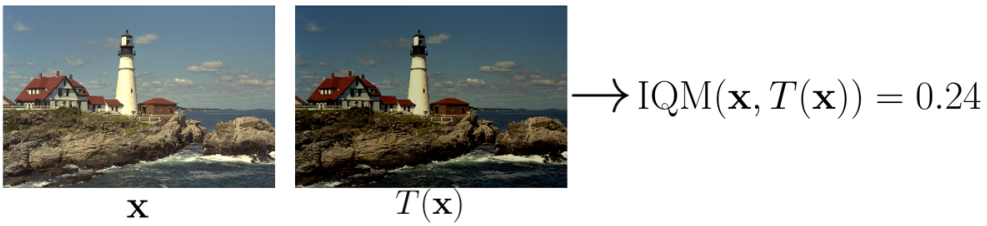

Figure 1: Example of a full reference image quality metric (IQM). A distortion is applied to image $\mathbf{x}$ to transform it into $T(\mathbf{x})$. The IQM outputs an objective measure of the difference between the two images with a scalar value. The Mean Squared Error (MSE) is shown shown as an example of an IQM with a scalar value of 0.24.

There are a plethora of IQMs to choose, the simplest being distances in Euclidean space such as mean squared error. In traditional perceptual literature, IQMs can be categorised into two groups. The first group operates on the premise that the image's structure remains unchanged despite the presence of distortion, adhering to the principle of *structural similarity*. The best example of this kind is the SSIM index [17]. The second group aims to measure *visibility of error* by quantifying how much distortion is visible to humans. Traditional IQMs in this group model the human visual system, converting images to a perceptually meaningful space in which a Euclidean distance is computed. Recent literature utilises deep learning models in an attempt to mimic human perception by correlating the model's response with image quality ratings from human experiments [4, 19, 6]. Figure 2 shows a comparison of IQMs for different distortions.

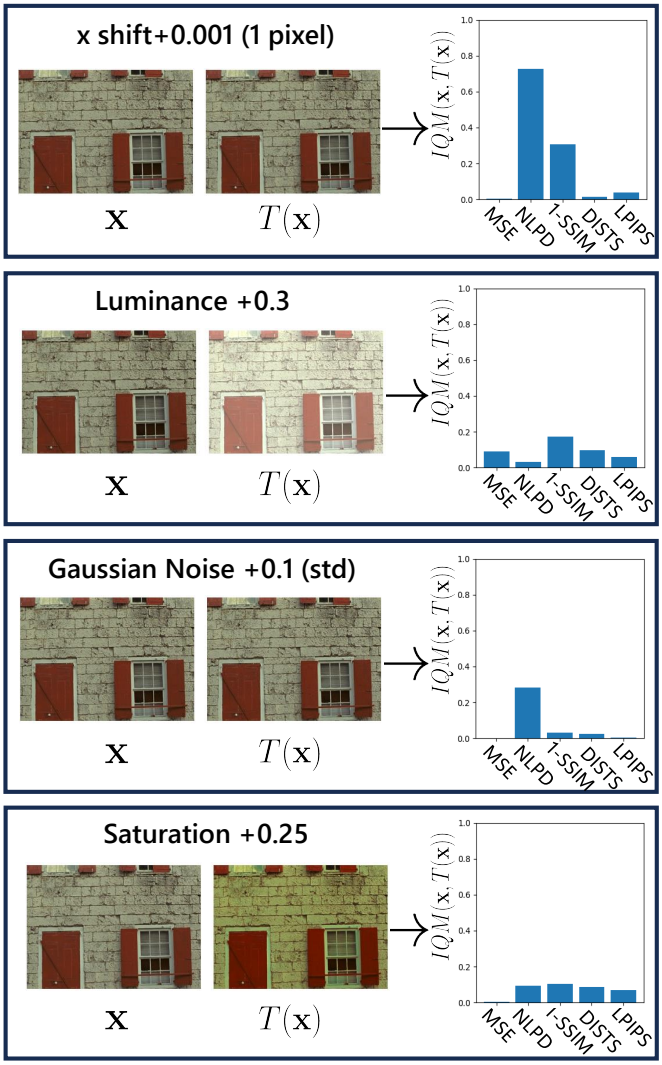

Figure 2: Comparing IQM scores for different distortions. DISTS [4] is designed to be invariant to small spacial perturbations as shown in the top row with X shifting the image. NLPD [10] is most sensitive to X shifting and Gaussian noise. SSIM [17] is sensitive to X shifting, luminance and saturation. MSE is only sensitive to luminance. LPIPS [19] is not sensitive to this level of Gaussian noise.

1.1 Comparing IQMs

The process of evaluating IQMs is both qualitative and quantitative. It is necessary to gather empirical data on the response profiles of different IQMs over specific image distortions and parameter ranges. Our toolbox, IQM-Vis, provides this functionality. An example is shown in Figure 3 where we can argue that: SSIM's response to changes in brightness is asymmetric, NLPD is more sensitive to saturation and Gaussian noise, and that SSIM is more sensitive to negative brightness. Furthermore we can deduce that beyond a value of $\sigma = 0.35$ Gaussian noise has no significant effect on the IQMs.

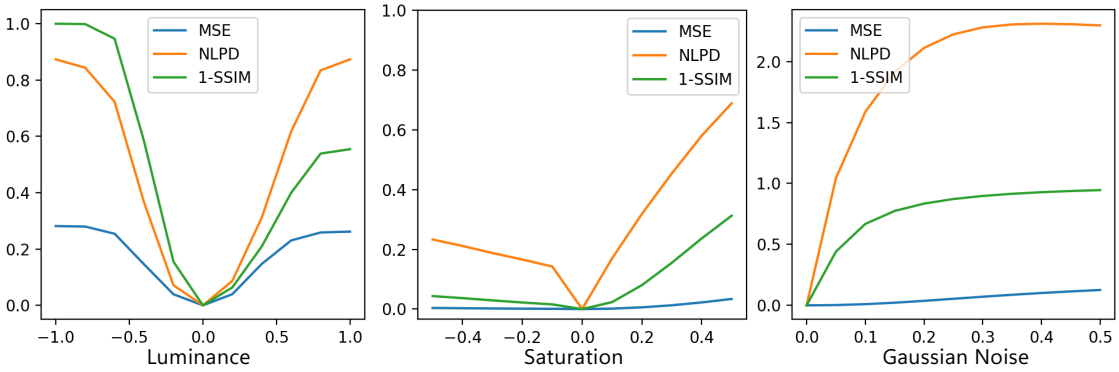

Figure 3: Quantitative comparison of IQM response profiles over a parameter range for three different distortions. Examples of the images used are shown in Figure 4.

It is important for a human to oversee the images produced from the distortion process to give context to the image data, as shown in Figure 4. This ensures that the desired qualities of an image distortion are present for a specific image sample.

Figure 4: Images transformed by multiple distortions. Visualising distortions over a range of parameter values helps to understand their effects on image quality. The top row corresponds to the images used for the saturation plot in Fig. 3.

Practitioners may require specific behaviour from an IQM which is not observable by measuring correlation over a dataset between human ratings and distances given by the IQM (Figure 7). For example, analysing ad-hoc distortions such as affine transformations is often extremely important (Figure 5). Evaluating specific desirable behaviour of IQMs, such as difference maps, lends itself best to *interactive* and *graphical* software.

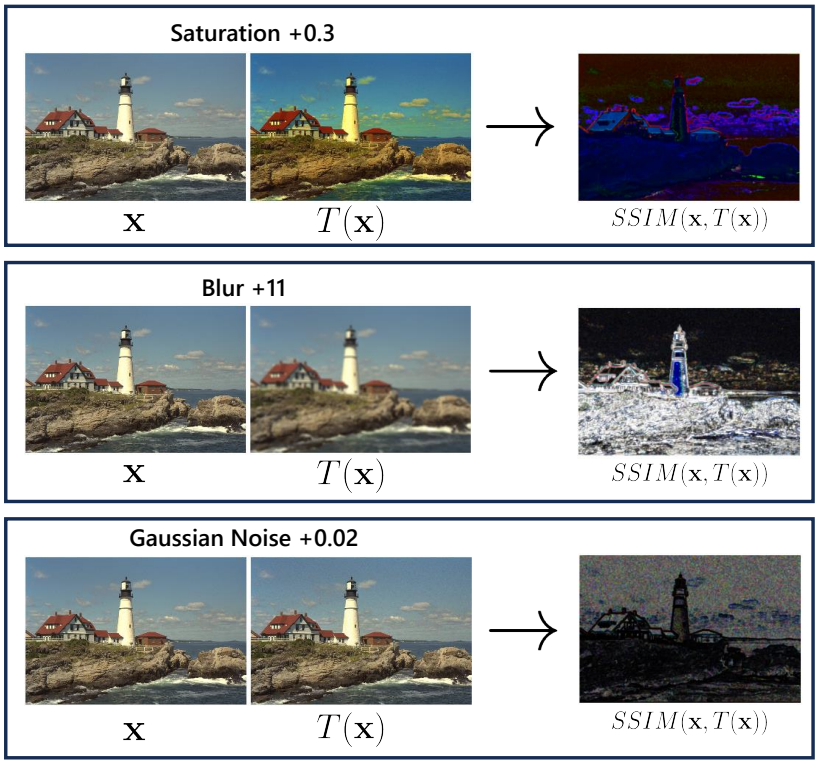

Figure 5: SSIM [17] difference maps. The mean of the difference map is taken to produce a scalar score, but it is useful to study the difference maps themselves. Note that SSIM is sensitive to different areas of the image for different distortions. Saturation: house and clouds. Blur: rocks and sea. Gaussian noise: sky.

2. Software Description

IQM-Vis is a toolbox which improves the quality and timescale of the IQM evaluation process. It enables quick access to visualising image distortions and evaluating IQMs through a simple and convenient Python graphical interface. Many standard distortions and IQMs are included [4, 10, 17, 19], and adding custom distortions, IQMs and image datasets is straightforward. Comparison graphs of IQMs are automatically generated as well as the option to compare with human scores to understand failures of an IQM with particular distortions or images. IQM-Vis is also able to perform two alternate force choice (2AFC) experiments, popular in the perceptual literature [14, 13, 19]. The package manages the data storage from the experiments and shows the results with correlation plots against desired IQMs. Any image distortions which do not conform to the correlation can be selected for further analysis of the image properties and visually inspected by the user.

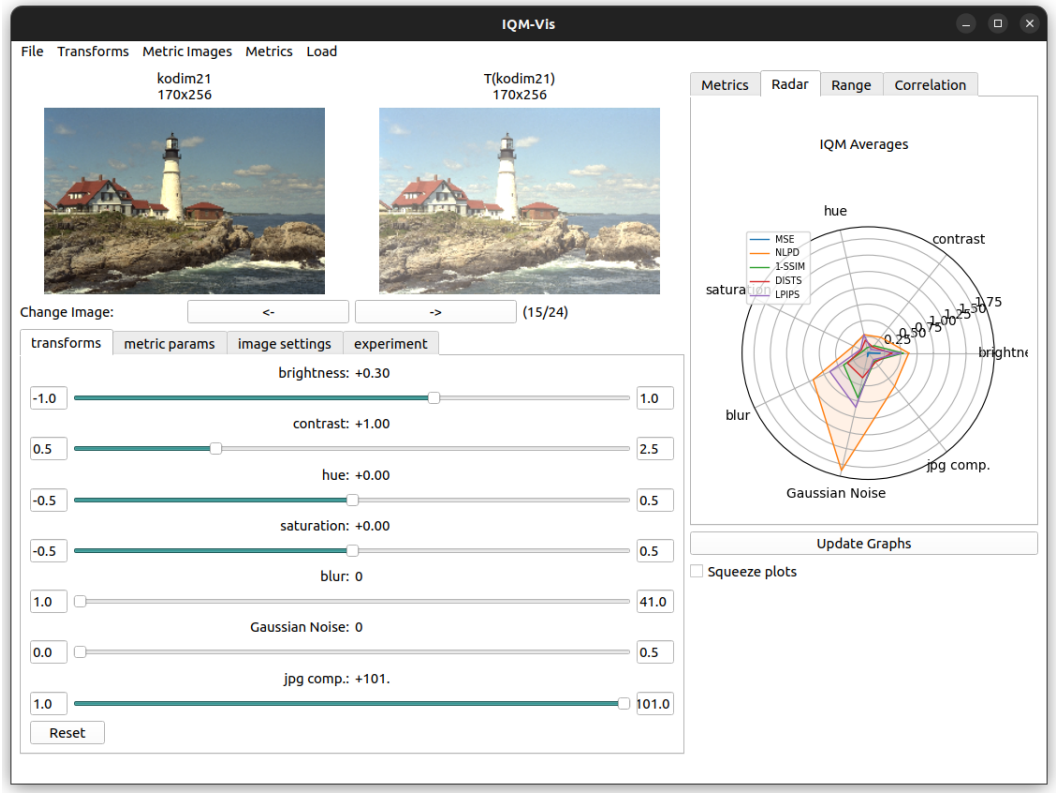

Figure 6: Screenshot of the IQM-Vis UI. Top left shows the reference and distorted images (Fig. 4), bottom left shows the distortions parameters. The 'metric params' option allows a user to change parameter values when tuning or designing IQMs. The 'image settings' option allows for image and screen calibration. Additional IQM and distortions can be selected via the drop down menu bars. A choice of IQM comparison graphs can be viewed on the right: radar, bar (Fig. 2), range (Fig. 3) and correlation (Fig. 7). Metric images (Fig. 5) can be viewed when selected from the drop down menu bar.

2.1 Quantitative Analysis

IQM-Vis provides a comprehensive analysis of IQMs through various quantitative graphs. This offers practitioners a well-rounded understanding of how different IQMs perform under differing scenarios. Figure 3 shows an example of range plots produced from IQM-Vis. These compare response profiles of IQMs over the range of a distortion's parameters, showing specifically what an IQM is sensitive against. On the right hand side of Figure 6 the a radar plot shows the mean score of each IQM over all parameter values of a distortion. This is useful to give an overarching view of which IQMs are sensitive to each distortion. Where data is available, users can view correlations to human perception scores for each IQM (Figure 7).

2.2 Qualitative Analysis

IQM-Vis provides multiple ways to inspect IQM and image distortion data. As shown in Figure 4, distortion effects can be viewed by adjusting their parameter slider. IQM scores are displayed under the metrics graph tab as shown in Figure 2. IQM difference maps (Figure 5) are displayed in the UI by selecting them in the drop down menu. When investigating correlation to human perception scores, any specific image sample which breaks the correlation can be quickly examined by clicking on the data point. This is useful for understanding where breakdowns in correlations occur.

2.3 Human Perception Experiments

Since most IQMs aim to capture an understanding of how humans perceive images, it is necessary to evaluate IQMs against image quality rating from humans. Popular datasets of human scoring exist such as TID[14, 13],

KADID-10K[11], and BAPPS[19]. However, these are static datasets using a subset of natural images and distortions. When designing and evaluating IQMs for use cases outside of these datasets, it is important to run human perceptual experiments to gather supplementary data. This ensures that the robustness and limitations of IQMs are understood and extends beyond the scope of pre-existing datasets.

IQM-Vis contains the functionality to conduct human perception experiments where participants provide a ranking of the perceived quality of an image exposed to specific distortions. The ranking is determined by a two alternative forced choice (2AFC) test which is a standard procedure for IQM datasets [19, 14, 13]. The 2AFC asks a test participant to choose out of two distorted images, which is most similar to a reference. The 2AFC is repeated until the underlying sorting algorithm (quick sort) provides an ordered list, s, of the distorted images rated by the participants from most similar to the reference getting progressively less similar. A human score is then associated with a given distorted image, $T_i(\mathbf{x})$ by

$$\text{score(i)} = \frac{s(i)}{|s|} \tag{1}$$

where $s(i)$ is the position of image $T_i(\mathbf{x})$ in the ordered list s and $|s|$ is the total number of images in experiment.

Experimentation enables practitioners to analyse custom distortions and image datasets. This is crucial for real-world design and evaluation of IQMs. Integrating the capacity to conduct experiments into IQM-Vis lowers the barriers for IQM research and analysis to beyond computer science experts. IQM-Vis facilitates the management and analysis of data collected through correlation graphs and statistics. Simple csv data structures are stored for additional out of application analysis and sharing. Importantly, IQM-Vis offers display image calibration for size and luminance to ensure consistency across experimental configurations. An overview of the steps to conduct an experiments are shown in Figure 7 and are exemplified in our demonstration video.

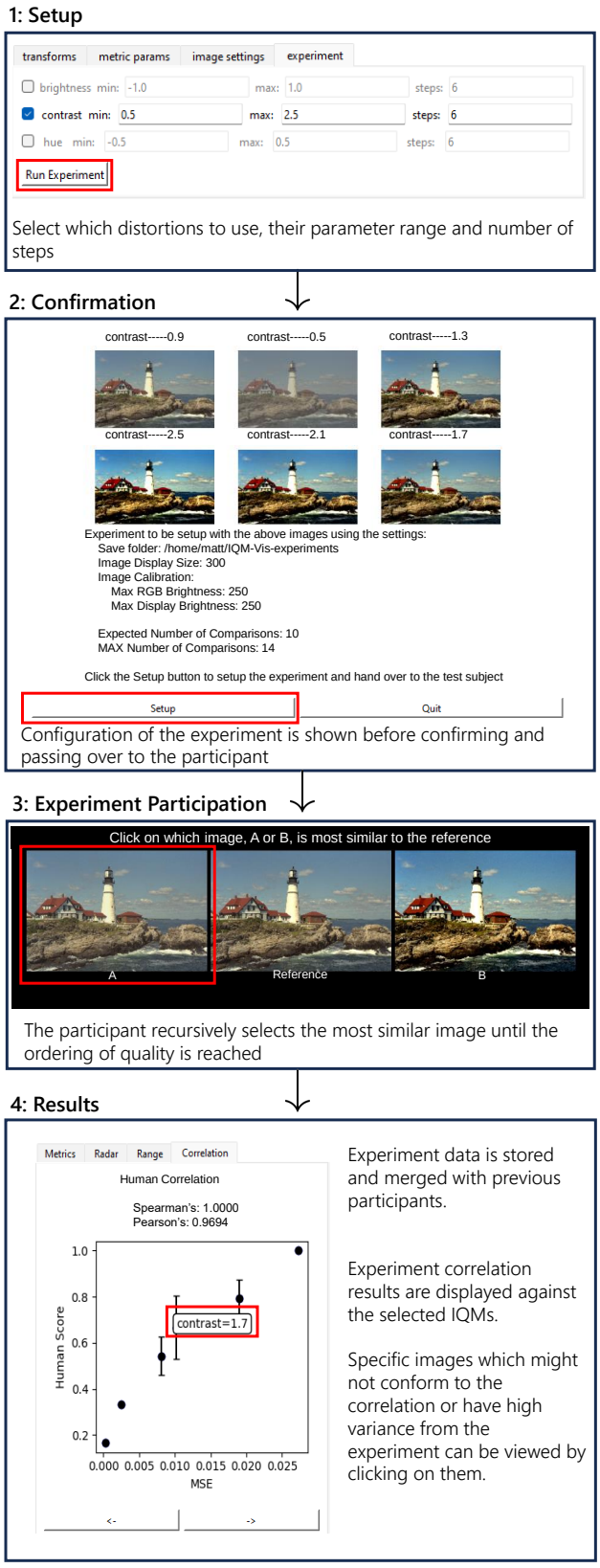

Figure 7: Running a human perception experiment in IQM-Vis. Each step is a cropped screenshot from the IQM-Vis interface. Red boxes indicate where the user clicks to progress each step.

2.4 Installation and usage

IQM-Vis requires python 3.9 or above and has several scientific libraries as dependencies which can be found in the `requirements.txt` file of the package.

A PyPi repository is maintained for ease of downloading and installation. Detailed instructions of how to install IQM-Vis are found in the getting started page of the documentation.

2.5 Customisation

IQM-Vis is customisable through a Python interface. Custom IQMs, distortions and parameters follow a simple API. All of the UI, graph generation, data storage and optimisation caching is handled by IQM-Vis. An example of simple IQM functions or image distortions is shown in Figure 8 and is explained in detail in our online tutorial.

Figure 8: Python code format for a user to add their own custom IQM or image distortion/transformation with IQM-Vis. Left: defining a simple IQM. Right: defining a simple distortion. More detailed examples can be found in our tutorials.

2.6 Support material

IQM-Vis is implemented in Python, the graphical user interface (GUI) is developed using PyQt6 for cross-platform compatibility, the image handling and processing is covered by OpenCV [2]. Numpy [5] is used for image distortions and other backend tasks and many of the IQMs provided take advantage of the PyTorch [12] framework to enable GPU hardware acceleration. Other software used is: pandas [18] for database management, matplotlib [9] for plotting graphs, SciPy [16], scikit-image [15].

To accompany the software, we have prepared online documentation, videos and tutorials that provide step-by-step instructions on installing, running and customising IQM-Vis.

3. Software Impacts

To the best of our knowledge, IQM-Vis is the only dedicated application and toolbox for both quantitative and qualitative analysis of IQMs. This is essential to understanding IQM's behaviour and suitability for a given use case.

We debuted IQM-Vis in 2023 at the International Conference on Computer Vision (ICCV) on the demos track, where there was significant interest from researchers and industry. In particular, the custom dataset creation described in Section 2.3, which lowers the boundaries to research on human perception by removing the overhead of explicitly programming 2AFC experiments which requires GUI, algorithmic and data management expertise. This opens the door to researchers with a non computer science background to conduct experiments on IQMs with different properties from vision science. For example, IQM-Vis allows experimental psychologists to quickly and easily design simple tests for IQMs in order to test a hypothesis, for example invariance to rotations.

IQM-Vis is presently integrated into the curriculum of the Masters of Artificial Intelligence program at the BME institute. Where within one of the modules, students acquire knowledge of various techniques for image comparison. As part of the learning process, students engage in a brief self-experiment to assess disparities in images and juxtapose their perceptions with diverse IQMs. IQM-Vis assists students in comprehending the distinct responses of various IQMs to image modifications and facilitates a comparison with their own subjective perceptions.

We envisage that the current field of research into model invariances or sensitivities [3, 1] will benefit greatly from IQM-Vis. For example, a user can conduct an experiment to discover how sensitive humans are to affine

transforms, and determine if an IQM designed to be invariant is actually invariant to the transformations. This enables researchers to target the design of IQMs with greater effect as well as enabling them to verify properties of IQMs as described in the literature.

More generally, IQM-Vis will greatly influence the evaluation of IQMs, which until now has strictly focused on correlations within existing perceptual datasets. IQM-Vis will be used in the design and tuning process of IQMs, allowing practitioners to continuously modify their design and parameter choices according to the feedback provided by IQM-Vis.

4. Conclusion

IQM-Vis is an open source toolbox to facilitate the analysis and understanding of IQMs. It provides the graphical interface, data storage and algorithmic knowledge required for thorough investigations of the properties of existing and novel IQMs. The software package includes a collection standard distortions and state-of-the-art IQMs, whilst still being easily customisable to a practitioner's requirements.

The graphical interface allows for both quantitative and qualitative analysis to be simultaneously undertaken. This is crucial when studying images for human consumption and algorithms which aim to mimic aspects of human perception. Inbuilt functionality to conduct 2AFC human perception experiments provides researchers with the tools to create custom datasets and explore IQM performance in areas of image distributions beyond what is currently available within the literature.

The toolbox is modular in nature in order to easily grow the set of available distortions and IQMs. End users have been involved in the co-design of the toolbox, which is a principle that will follow into future versions of IQM-Vis as the user base's demands and needs evolve.

Acknowledgements
This work is supported by the UKRI Centre for Doctoral Training in Interactive AI EP/S022937/1 and the UKRI Turing AI Fellowship EP/V024817/1.

References

[1] Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew G Wilson. Learning invariances in neural networks from training data. *Advances in neural information processing systems*, 33:17605–17616, 2020.

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] Valentin Delchevalerie, Adrien Bibal, Benoît Frénay, and Alexandre Mayer. Achieving rotational invariance with bessel-convolutional neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 28772–28783. Curran Associates, Inc., 2021.

[4] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020.

[5] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

[6] Alexander Hepburn, Valero Laparra, Jesús Malo, Ryan McConville, and Raul Santos-Rodriguez. Perceptnet: A human visual system inspired neural network for estimating perceptual distance. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 121–125, 2020.

[7] Alexander Hepburn, Valero Laparra, Ryan McConville, and Raul Santos-Rodriguez. Enforcing perceptual consistency on generative adversarial networks by using the normalised laplacian pyramid distance. In *Proceedings of the Northern Lights Deep Learning Workshop*, volume 1, pages 6–6, 2020.

[8] Alexander Hepburn, Valero Laparra, Raul Santos-Rodriguez, Johannes Ballé, and Jesus Malo. On the relation between statistical learning and perceptual distances. In *International Conference on Learning Representations*, 2022.

[9] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[10] Valero Laparra, Alexander Berardino, Johannes Ballé, and Eero P Simoncelli. Perceptually optimized image rendering. *JOSA A*, 34(9):1511–1525, 2017.

[11] Hanhe Lin, Vlad Hosu, and Dietmar Saupe. Kadid-10k: A large-scale artificially distorted iqa database. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3. IEEE, 2019.

[12] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[13] Nikolay Ponomarenko, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, et al. Image database tid2013: Peculiarities, results and perspectives. *Signal processing: Image communication*, 30:57–77, 2015.

[14] Nikolay Ponomarenko, Vladimir Lukin, Alexander Zelensky, Karen Egiazarian, Marco Carli, and Federica Battisti. Tid2008-a database for evaluation of full-reference visual quality assessment metrics. *Advances of modern radioelectronics*, 10(4):30–45, 2009.

[15] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.

[16] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[17] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[18] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.

[19] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.