

Article

# Blind Image Quality Assessment with Deep Learning: A Replicability Study and Its Reproducibility in Lifelogging

Ricardo Ribeiro , Alina Trifan  and António J. R. Neves \* 

IEETA/DETI, University of Aveiro, 3810-193 Aveiro, Portugal

\* Correspondence: an@ua.pt

**Abstract:** The wide availability and small size of different types of sensors have allowed for the acquisition of a huge amount of data about a person's life in real time. With these data, usually denoted as lifelog data, we can analyze and understand personal experiences and behaviors. Most of the lifelog research has explored the use of visual data. However, a considerable amount of these images or videos are affected by different types of degradation or noise due to the non-controlled acquisition process. Image Quality Assessment can play an essential role in lifelog research to deal with these data. We present in this paper a twofold study on the topic of blind image quality assessment. On the one hand, we explore the replication of the training process of a state-of-the-art deep learning model for blind image quality assessment in the wild. On the other hand, we present evidence that blind image quality assessment is an important pre-processing step to be further explored in the context of information retrieval in lifelogging applications. We consider that our efforts have been successful in the replication of the model training process, achieving similar results of inference when compared to the original version, while acknowledging a fair number of assumptions that we had to consider. Moreover, these assumptions motivated an extensive additional analysis that led to significant insights on the influence of both batch size and loss functions when training deep learning models in this context. We include preliminary results of the replicated model on a lifelogging dataset, as a potential reproducibility aspect to be considered.

**Keywords:** image processing; blind image quality assessment; deep learning; replicability; reproducibility; image retrieval; lifelogging



**Citation:** Ribeiro, R.; Trifan, A.; Neves, A.J.R. Blind Image Quality Assessment with Deep Learning: A Replicability Study and Its Reproducibility in Lifelogging. *Appl. Sci.* **2023**, *13*, 59. <https://doi.org/10.3390/app13010059>

Academic Editor: Jan Egger

Received: 28 November 2022

Revised: 14 December 2022

Accepted: 15 December 2022

Published: 21 December 2022



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The exponential growing trend in the amount of digital data generated over the last decade has been accompanied by the increase in attention from multiple research communities (e.g., signal processing, computer vision, and information retrieval, etc.). With respect to the computer vision community, continuous efforts are focused on the development of imaging systems able to capture, process, store, compress, transmit, and display digital images. One specific challenge still to be overcome by these systems relates to the fact that digital images can be affected by a wide range of degradation, also denoted in related literature as distortion, namely noise, JPEG artifacts, aliasing, lens and motion blur, over-sharpening, wrong exposure, color fringing, and over-saturation, among others. This, more often than not, results in a degradation of their visual quality, which affects the remaining building blocks of such architectures [1].

Image Quality Assessment (IQA) is a research area that focuses on the development of computational models that automatically measure the perceptual quality of an image. Such models are needed to evaluate and improve the quality of digital images without human intervention. IQA methods have been presented in the literature in three main categories [2]:

- Full-reference (FR) to predict quality scores of a target image with full access to the original reference image;

- Reduced-reference (RR) to predict quality scores of a target image with limited access to the reference image;
- No-reference (NR) or Blind to predict quality scores of a target image without any information about original reference image.

The perceptual quality score is a subjective measure. It is generally defined as the mean of the individual ratings of perceived quality assigned by human subjects, and it is also called Mean Opinion Score (MOS) [3]. Several FR and RR approaches have been proposed in the literature with high performance on the task of assessing image quality [4,5]. However, when it comes to real-world scenarios, these methods are negatively impacted by the non-availability of the reference images. Without having access to a reference image, developing accurate quality assessment methods becomes more challenging. Current research efforts in Blind IQA (BIQA) methods concentrate on advancing robust solutions for image quality assessment, with applications in areas where having access to a reference image at the time of the assessment is not possible.

Recently, deep learning approaches developed for image classification tasks have outperformed traditional methods, which has attracted the attention of researchers in the field of IQA. Consequently, deep learning methods have been developed for BIQA purposes, mainly as a consequence of the ability of Convolutional Neural Networks (CNN) to extract powerful features from images without the need of having a reference image [6,7]. BIQA algorithms are usually categorized into two classes according to the distortion types [1,2] they target:

- Distortion-specific algorithms [8], which use specific distortion models to predict the perceptual quality and measure one or more distortions in an image;
- General-purpose algorithms [9,10], which measure the perceptual quality of an image across several types of distortions by using the general quality features implied in the image.

We have identified general-purpose BIQA methods as having great potential for applications in unconstrained environments, such as lifelogging applications. Lifelogging is the process of tracking and recording personal data created through our activities and behavior [11]. In particular, this personal data, also called lifelogs, can be captured in the form of visual lifelogs such as images or videos [12]. Information retrieval is an important aspect connected to lifelogging since in most lifelogging applications the objective is to accurately retrieve lifelogs that answers to a specific research question. Therefore, we consider BIQA to be of fundamental interest for lifelogging applications, as not all the captured visual lifelogs (more specifically, images) have relevant information for the task in question. Several images captured in a lifelogging process are affected by different types of degradation, mostly due to the use of egocentric camera devices. By filtering out such images using BIQA approaches, we are able to ignore images with low visual impact for the lifelogger, saving processing time and increasing the performance of the system when retrieving the desired visual lifelogs. Moreover, using this filtering process, we do not try to interpret degraded images which could lead to poor annotations, due to the fact that most of the computer vision tasks, such as image annotation, activity recognition, face recognition, and object segmentation, among others, are trained with acceptable quality images. The annotations are more accurate with a positive impact on the retrieval phase, mostly performed by text queries.

The primary goal of our work is a replicability study of a BIQA deep learning methodology (KonCept512) developed and documented by Hosu et al. [9]. Secondly, we are interested in reproducing their work in a lifelogging application [13]. The authors of the study that we replicate in this manuscript proposed a large IQA in-the-wild dataset (KonIQ-10k) (<http://database.mmsp-kn.de/koniq-10k-database.html>, accessed on 25 November 2022) to evaluate and benchmark IQA algorithms with regard to the authenticity of degradation, the diversity of content, and quality-related indicators. We are interested in reproducing their results on a different, lifelogging dataset. In summary, our work advances the following contributions:

- Replicability of the best model trained by Hosu et al. [9], KonCept512.
- If the replicability is possible, we will evaluate the training of the best model (KonCept512) under different hardware setup.
- Reproducibility of the best model documented by Hosu et al. [9] on a lifelogging dataset.
- Discussion of the replicability and reproducibility journeys and lessons learned.

This remainder of this paper is organized as follows: In Section 2, we provide a description of the best model proposed by Hosu et al. [9], KonCept512. The experimental setup and the procedures conducted in this paper are presented in Section 3. In Section 4, we present the obtained experimental results. We were able to replicate the original work up to a certain extent. We compare the performance of the model trained under different conditions, and we detail our reasoning for some of the decisions that were taken during the process. Moreover, we address the reproducibility issue through an analysis of the MOS distribution in a lifelogging dataset. Finally, we reflect on the lessons learned in Section 5.

## 2. KonCept512 Model

The original BIQA manuscript [9] presented several approaches followed in order to find a better end-to-end deep BIQA architecture. These approaches took into consideration three main factors: CNN base architectures proposed for image classification (ImageNet [14]), several loss functions, and different input sizes and combinations of the above factors. The authors found that the best performing model (named KonCept512) was obtained by applying the InceptionResNetV2 base architecture [15], with a Mean Square Error (MSE) loss, that was trained and tested on images of input size of  $512 \times 384$ .

We explored KonCept512 model in this work. Figure 1 shows the architecture proposed by Hosu et al. [9], in which the KonCept512 model was trained. The input image is passed through the InceptionResNetV2 backbone (without the final fully-connected layers) pre-trained on ImageNet dataset in order to extract features from the image, followed by a Global Average Pooling (GAP) layer. These layers are connected to four Fully-Connected (FC) layers. The first three layers have 2048, 1024, and 256 units, respectively. Each one of these FC layers are followed by an activation function, the Rectified Linear Unit (ReLU), and a dropout layer, each with rates of 0.25, 0.25, and 0.5 in order to avoid over-fitting, followed by a final linear FC layer. This output layer has one output unit to predict MOS.



**Figure 1.** End-to-end architecture of trained KonCept512 model to predict MOS values [9].

## 3. Experimental Setup

We initiated this study by following the methodology and the source code presented by the authors at [16]. Two relevant aspects had to be embraced from early on, in order for us to be able to conduct the study. First, the code available at (<https://github.com/subpic/koniq>, accessed on 25 November 2022) was implemented by the authors on Google Colab notebooks (<https://colab.research.google.com/>, accessed on 25 November 2022). In a first attempt, we tried to run the code on a Google Colab platform, under a free plan. However, the resources of this platform are not guaranteed as static allocations and are limited under these conditions. As such, we were not able to simply execute the code as is. A first straightforward conclusion was that in order to be able to train the KonCept512 model using Google Colab infrastructures, we would have to set up a paid plan. This financial constraint, which was not explicitly addressed in the original manuscript, led us to the use of local computing infrastructure for the remainder of the study.

Second, the authors in [9] used Python Keras library (<https://keras.io/>, accessed on 25 November 2022) with TensorFlow (<https://www.tensorflow.org/>, accessed on 25 November 2022) as a backend and ran on two NVIDIA Titan Xp GPUs (<https://www.nvidia.com/en-us/titan/titan-xp/>, accessed on 25 November 2022). Nevertheless, they do not specify the versions of the library used. As a result, we had to make several assumptions, namely batch size and library versions, and adjust the code in order to replicate the author's code on our local infrastructure (one NVIDIA RTX 3080 ti GPU (<https://www.nvidia.com/en-eu/geforce/graphics-cards/30-series/rtx-3080-3080ti/>, accessed on 25 November 2022)).

This section details all the steps explored during the experimental setup that led us to results that are very similar to the ones presented by the authors in [9]. Moreover, as previously introduced, we applied the Koncept512 models to a lifelogging dataset with the objective of predicting the MOS of thousands of images captured in a person's daily life.

### 3.1. Datasets

Hosu et al. [9] evaluated the proposed model on two benchmark datasets: the one collected and introduced by the authors in the same paper (KonIQ-10k) and the LIVE in the Wild (LIVE-itW) [17] dataset. Hosu et al. [9] divided the KonIQ-10k dataset into three sets: training (7058 images), validation set (1000 images), and test (2015 images). The validation set was used to evaluate the best generalizing of their best performing model, Koncept512. The final model evaluation was performed on the test set, and it was also evaluated on the test partition of the LIVE-itW dataset. The images for each set were selected by the previous authors.

For the experiments we pursued with respect to reproducing their work in a lifelogging setup, we employed the dataset of the fourth Lifelog Search Challenge (LSC'21) [18]. This dataset contains more than 180 thousand wearable camera images captured over four mouths from an active lifelogger.

### 3.2. Evaluation Metrics

The performance of BIQA algorithms are mainly assessed based on two metrics [1,2,9]: the Spearman Rank-order Correlation Coefficient (SRCC) Equation (1) and the Pearson Linear Correlation Coefficient (PLCC) Equation (2).

$$SRCC = 1 - \frac{6 \sum_i^N d_i}{N \times (N^2 - 1)}, \quad (1)$$

where  $d_i$  is the difference between the  $i$ th image ranks in subjective and objective evaluations in an image database of size  $N$ . SRCC can assess the prediction monotonicity of an BIQA method [1]:

$$PLCC = \frac{\sum_i (q_i - \bar{q}) \times (o_i - \bar{o})}{\sqrt{\sum_i (q_i - \bar{q})^2 \times (o_i - \bar{o})^2}} \quad (2)$$

where  $o_i$  is the subjective score (MOS) of the  $i$ th image, and  $q_i$  is the predicted objective score.

### 3.3. Replicability Pipeline

As previously stated, due to the limited resources of a Google Colab free plan, we were not able to replicate the Koncept512 model using the exact same experimental setup of the original authors in [16]. Consequently, we adjusted the provided code for Python3 ([www.python.org](http://www.python.org), accessed on 25 November 2022) Google Colab notebook on our local machine. We updated the code to run with the most recent version of Python (3.8) and TensorFlow (2.6) that includes the Keras API. Unlike the setup used in the original paper, our local machine only provides one GPU (RTX 3080ti) with the latest CUDA Toolkit (11.4) (<https://developer.nvidia.com/cuda-toolkit>, accessed on 25 November 2022) and GPU-accelerated libraries versions. As the authors do not provide a requirements list with all the needed libraries to run on a local machine, we performed an analysis of the code in

order to identify them. We then proceeded to installing their most recent stable versions on the local machine.

In the training process, a custom learning rate was applied by first training the training set for 40 epochs with a learning rate of  $10^{-4}$ , monitoring the PLCC on the validation set, and saving the best performing model. After training these 40 epochs, the best model is loaded and trained for another 20 epochs with lower learning rate (of  $5 \times 10^{-5}$ ). The authors then follow the same procedure for another 10 epochs with learning rate of  $10^{-5}$ . They do not specify the batch size for the KonCept512 model; however, they used the largest batch size images that fit on the available GPU memory in [9], so we inferred that the batch size would be 16. However, our GPU architecture can not handle a batch size images of 16. We used this apparent limitation as a drive for understanding the effect of the batch size in the training phase. We trained the model three times, with different batch sizes (of 2, 4 and 8).

We made another experiment by changing the loss function. In the training process, we trained a model by maximizing the PLCC with a loss functions based on the PLCC metric instead of MSE loss function. MSE loss is a loss function widely-used in BIQA to minimize the error of the trained model. However, in recent studies [19–21] a PLCC-induced loss has been applied together with other loss functions in order to train models. By using these type of loss functions the models could converge faster and better than with MSE loss [20]. After training this model, a model was trained by decreasing the batch size from 8 to 4, to understand the impact of the batch size in case of using a PLCC-induced loss function.

It is important to note that we contacted the authors and asked them to clarify the computations that were performed in the validation stage. Their response was quick and detailed. We also requested for an implementation of the code that would not rely on Google Colab. The authors replied with an updated version of the code, with minor adjustments in terms of libraries versions. However, the updated code also needed to be running within Google Colab and the authors stated they have no experience with running the code outside these computing infrastructures. The updated code that the authors shared were consistent with the initial assumptions we made in order to be able to run the code on our local machine. Nonetheless, we trained two more models using the updated libraries, exactly as proposed by the authors. We followed their code by training the models following the same procedure as the first trained. It should be noted that these models were not trained with same number of epochs as described in [9], instead the authors excluded the final 10 epochs with learning rate of  $10^{-5}$  of the training process.

As we have previous experience on training deep learning models with PyTorch, a machine learning framework similar to Tensorflow, and taking into consideration the initial conflicts of libraries versions using Tensorflow, we found a PyTorch reproduction of the original code to train the KonCept512 model (<https://github.com/ZhengyuZhao/koniq-PyTorch>, accessed on 25 November 2022). We made some adjustments in the PyTorch to train and test the KonCept512 model in our local machine. So in addition to the models already mentioned, we also trained a model with the implementation in PyTorch following the same procedure mentioned above with a batch size of 8, that fits on our GPU memory. Moreover, in our lifelogging application MEMORIA [13], we implemented all the deep learning modules for image processing in PyTorch. Finally, we tested all the trained models in the test set of the two IQA datasets mentioned in Section 3.1.

For the reproducibility step, we applied the original pre-trained Koncept512 model proposed by Hosu et al. [9] and our comparative model to the lifelogging dataset presented in Section 3.1, while the obtained results of this step are preliminary, we were interested in understanding the capability of these models in the lifelogging arena, where image quality assessment could be a foundational step for a more accurate and time efficient image retrieval.

#### 4. Experimental Results

The results we obtained are not fully consistent with the results presented by the original authors. We believe this is mainly related to the physical setup that was used for the training process. More specifically, the setup impacted the maximum batch size that we could use in training. Consequently, the batch size affected the overall model performance, as we will detail next. Unlike the Google Colab infrastructure that the authors used in their original manuscript, our local machine could not fit the training process with a batch size greater than 8 (in the original paper a batch size of 16 was used). However, the results we obtained are close to the ones presented by the authors. We show results obtained with different batch sizes, which show its impact on the final model performance.

We explore the impact of batch size on the training process and compared the results with the original KonCept512 pre-trained model. The authors reflected in [9] that training some CNN base architectures with very small batch sizes and much larger input sizes limits the best possible performance that could be achieved. Table 1 shows the results obtained by testing the trained models with batch sizes of eight, four, and two on KonIQ-10k training set and testing these model on KonIQ-10k and LIVEitW test set. We compare these results to the original KonCept512 model trained in [9]. The results show that the smaller the batch size, the lower the performance of the model. Thus, our intuition would be that using a batch size of 16 and the procedures described in [9] with the assumptions we made in the beginning of this study, the original KonCept512 model could be reproduced with the same performance.

**Table 1.** Trained model with different batch size compared to the original pre-trained model KonCept512.

Model	Batch Size	Test on KonIQ-10k		Test on LIVE-itW	
		SROCC	PLCC	SROCC	PLCC
KonCept512 [9]	16	0.921	0.937	0.825	0.848
Current Study	8	0.889	0.903	0.800	0.809
	4	0.793	0.774	0.667	0.661
	2	0.426	0.413	0.363	0.349

In addition to these models, we explored the use of a loss function that reflect the maximization of the PLCC. The PLCC-induced loss function used is [20]:  $loss = (1 - plcc)/2$ , where  $plcc$  is the Pearson Linear Correlation Coefficient obtained from the training process. As the objective of the model is to achieve the best prediction linearity between the predicted perceptual quality scores and the subjective quality scores, the PLCC-induced loss could feature linearity [21]. Although this loss function is not used in isolation [19–21], we explored its impact on the training and testing of models. Table 2 shows the results in KonIQ-10k and LIVEitW test set obtained using the model trained in KonIQ-10k training set with the PLCC-induced loss function. Comparing the results in Table 1 with these results, we note that we obtained a small improvement in the performance of the models, even if we reduce the batch size to four. However, when we monitor the MSE values of the models, we observe that these values are much higher than models trained with MSE loss function. Nonetheless, in addition to the improvement of the evaluation metrics, we note that training models with MSE loss are more unstable when varying the batch sizes [20].

**Table 2.** Trained model with PLCC-induced loss function to maximize the PLCC with different batch size.

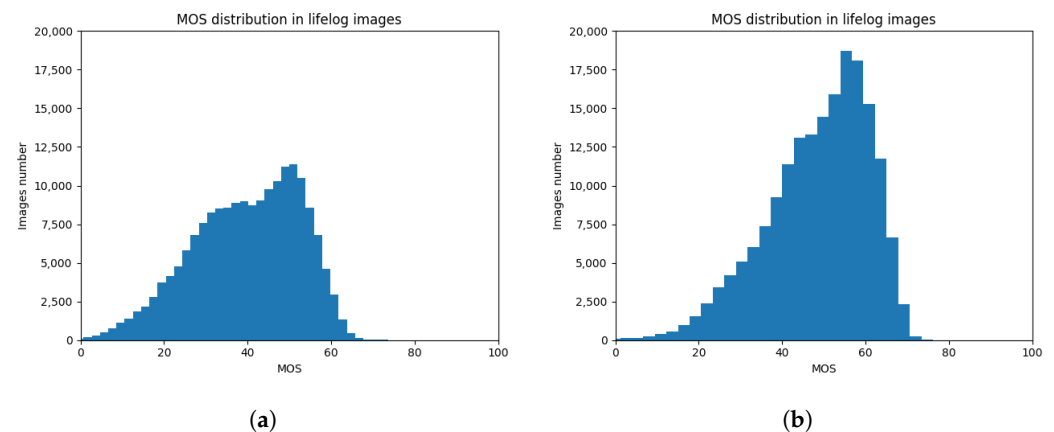
Loss Function	Batch Size	Test on KonIQ-10k		Test on LIVE-itW	
		SROCC	PLCC	SROCC	PLCC
PLCC	8	0.914	0.930	0.816	0.837
	4	0.871	0.877	0.748	0.713

Table 3 shows the results obtained from training a model with the updated libraries provided by the authors and the model trained by implementing the code using the PyTorch framework. As it can be seen, the results of these two models are very similar to the ones previously presented. By comparing these results to the original model KonCept512, we can conclude that by increasing the batch size to 16, whatever the framework used to train the models, we can achieve the same result as reported by Hosu et al. [9].

**Table 3.** Trained model with update version, and PyTorch version of authors compared to the original pre-trained model KonCept512.

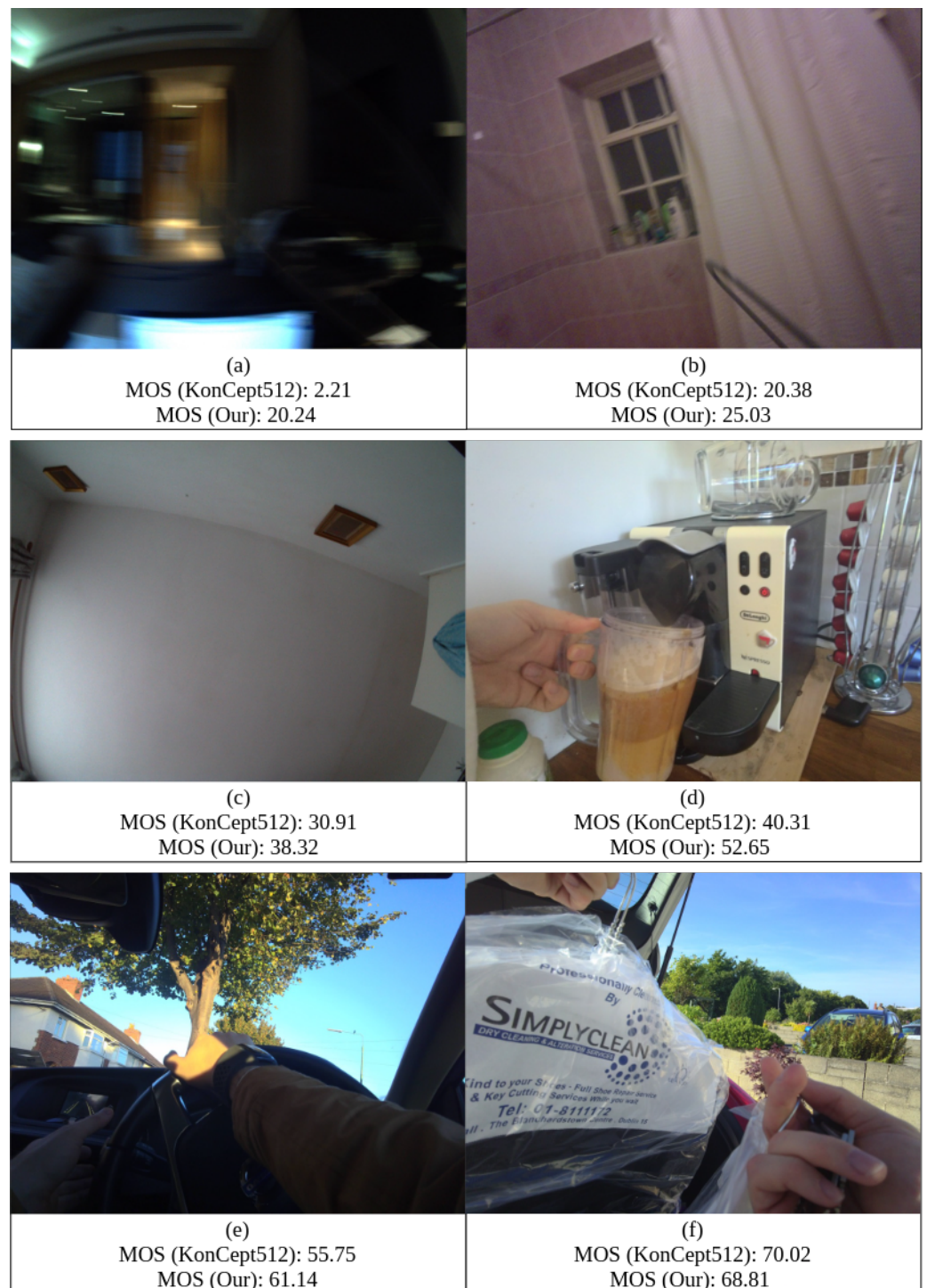
Model	Batch Size	Test on KonIQ-10k		Test on LIVE-itW	
		SROCC	PLCC	SROCC	PLCC
Koncept512 [9]	16	0.921	0.937	0.825	0.848
Current Study (Tensorflow)	8	0.902	0.921	0.796	0.803
Current Study (Pytorch)	8	0.892	0.913	0.783	0.805

As our final goal is to integrate a BIQA method into our lifelogging application [13], in order to filter images with low perceptual quality, we ran the original model KonCept512, as well as the model we trained in this reproducibility work, on a lifelogging dataset [18] previously introduced in Section 3.1. We predicted the MOS values of 183,227 images and analyzed the distribution of these images over the possible MOS values as shown in Figure 2. In Figure 3, we show several lifelog images with the respective MOS predicted using the original KonCept512 pre-trained model provided by Hosu et al. [9] and the model we trained using PyTorch.



**Figure 2.** MOS distribution predicted on a lifelog dataset using the pre-trained KonCept512 models: (a) provided by Hosu et al. [9] and (b) trained in this work.

By analyzing the histogram in Figure 2 and the examples shown in Figure 3, we hypothesize that if a MOS threshold is defined to filter the images, we are able to exclude thousands of irrelevant images and thus improve the image retrieval process in a lifelogging application. For example, considering a MOS threshold value of 30 in Figure 2a histogram, we would discard from processing a number of 42,913 images, which amounts to 25% of the dataset. We will further explore the use of BIQA models in lifelogging applications and also following the approach proposed by Hosu et al. [9], we will attempt to improve the results making changes on backbone architectures of the CNN, loss functions, just to name a few.



**Figure 3.** MOS prediction of images from lifelogging dataset [18].

The code produced in this work is available at <https://github.com/RRibeiro8/koniq-logger>, accessed on 25 November 2022.

## 5. Conclusions

We consider BIQA to be of fundamental interest for lifelogging applications, since several images captured in a lifelogging process are affected by different types of degradation, mostly due to the use of egocentric camera devices. We presented experimental results showing that after selecting a correct MOS threshold, we can discard from processing 25% of the images from the chosen dataset. By filtering out such images using BIQA approaches,



we are able to ignore images with low visually impact for the lifelogger, saving processing time and increasing the performance of the system when retrieving the desired visual lifelogs. Moreover, the annotations of the images in the lifelog data are more accurate with a positive impact on the retrieval phase, mostly conducted by text queries.

To achieve the previous conclusion, we presented in this paper a replicability study of a deep learning model designed for blind image quality assessment. Our experiments did not lead us to a full replication of the research contribution in question, but we obtained close inference results, while we recognize the thorough work and provided support by the authors of the original BIQA model, we have identified and discussed during the document several gaps that we tried to overcome to the best of our knowledge. These identified gaps allowed us to further infer significant knowledge that we hope could benefit a wide research community.

To summarize the main outcomes of the presented replicability study, we have to first refer to the physical resources that limited the exact replication of the previous work. Diving further into the lessons learned, we experimentally confirmed that a drawback of a smaller batch size is that the model is not guaranteed to converge to the global optima. However, it is well known in the machine learning community that the behavior of hyperparameters often varies from dataset to dataset and model to model. Therefore, the conclusions we make can only serve as references for the study we replicated. Finally, by experimenting with several loss functions, we can conclude that the choice of the loss function should consider the measure we want to optimize.

In future work, we will further explore the use of BIQA models in lifelogging applications, namely attempting to improve the obtained results making changes on backbone architectures of the CNN, exploring other loss functions, and use other training data more related to lifelog.

**Author Contributions:** Conceptualization, R.R., A.T. and A.J.R.N.; methodology, R.R., A.T. and A.J.R.N.; software, R.R.; writing—original draft preparation, R.R.; writing—review and editing, R.R., A.T. and A.J.R.N.; supervision, A.T. and A.J.R.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by National Funds through the FCT—Foundation for Science and Technology—in the context of the project UIDB/00127/2020.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. This data can be found here: <http://database.mmsp-kn.de/koniq-10k-database.html> or <https://doi.org/10.1109/TIP.2020.2967829>, accessed on 27 November 2022.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Xu, S.; Jiang, S.; Min, W. No-reference/blind image quality assessment: A survey. *IETE Tech. Rev.* **2017**, *34*, 223–245. [[CrossRef](#)]
2. Zhai, G.; Min, X. Perceptual image quality assessment: a survey. *Sci. China Inf. Sci.* **2020**, *63*, 211301. [[CrossRef](#)]
3. Leonardi, M.; Napoletano, P.; Schettini, R.; Rozza, A. No Reference, Opinion Unaware Image Quality Assessment by Anomaly Detection. *Sensors* **2021**, *21*, 994. [[CrossRef](#)] [[PubMed](#)]
4. Liu, T.J.; Liu, K.H.; Lin, J.Y.; Lin, W.; Kuo, C.C.J. A ParaBoost Method to Image Quality Assessment. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 107–121. [[CrossRef](#)] [[PubMed](#)]
5. Golestaneh, S.; Karam, L.J. Reduced-Reference Quality Assessment Based on the Entropy of DWT Coefficients of Locally Weighted Gradient Magnitudes. *IEEE Trans. Image Process.* **2016**, *25*, 5293–5303. [[CrossRef](#)] [[PubMed](#)]
6. Wu, J.; Ma, J.; Liang, F.; Dong, W.; Shi, G.; Lin, W. End-to-end blind image quality prediction with cascaded deep neural network. *IEEE Trans. Image Process.* **2020**, *29*, 7414–7426. [[CrossRef](#)]
7. Ma, J.; Wu, J.; Li, L.; Dong, W.; Xie, X.; Shi, G.; Lin, W. Blind Image Quality Assessment With Active Inference. *IEEE Trans. Image Process.* **2021**, *30*, 3650–3663. [[CrossRef](#)]

8. Min, X.; Zhai, G.; Gu, K.; Fang, Y.; Yang, X.; Wu, X.; Zhou, J.; Liu, X. Blind quality assessment of compressed images via pseudo structural similarity. In Proceedings of the 2016 IEEE International Conference on Multimedia and Expo (ICME), Seattle, WA, USA, 11–15 July 2016; pp. 1–6.
9. Hosu, V.; Lin, H.; Sziranyi, T.; Saupe, D. KoniQ-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE Trans. Image Process.* **2020**, *29*, 4041–4056. [[CrossRef](#)] [[PubMed](#)]
10. Su, Y.; Korhonen, J. Blind Natural Image Quality Prediction Using Convolutional Neural Networks And Weighted Spatial Pooling. In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 25–28 October 2020; pp. 191–195.
11. Gurrin, C.; Smeaton, A.F.; Doherty, A.R. Lifelogging: Personal big data. *Found. Trends Inf. Retr.* **2014**, *8*, 1–125. [[CrossRef](#)]
12. Ribeiro, R.; Trifan, A.; Neves, A.J. Lifelog Retrieval From Daily Digital Data: Narrative Review. *JMIR mHealth uHealth* **2022**, *10*, e30517. [[CrossRef](#)] [[PubMed](#)]
13. Ribeiro, R.; Trifan, A.; Neves, A.J. MEMORIA: A Memory Enhancement and MOment Retrieval Application for LSC 2022. In Proceedings of the 5th Annual on Lifelog Search Challenge, Newark, NJ, USA, 27–30 June 2022; pp. 8–13.
14. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
15. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
16. Hosu, V. Koniq. 2020. Available online: <https://github.com/subpic/koniq> (accessed on 25 November 2022).
17. Ghadiyaram, D.; Bovik, A.C. Massive online crowdsourced study of subjective and objective picture quality. *IEEE Trans. Image Process.* **2015**, *25*, 372–387. [[CrossRef](#)] [[PubMed](#)]
18. Gurrin, C.; Jónsson, B.T.; Schöffmann, K.; Dang-Nguyen, D.T.; Lokoč, J.; Tran, M.T.; Hürst, W.; Rossetto, L.; Healy, G. Introduction to the Fourth Annual Lifelog Search Challenge, LSC'21. In Proceedings of the International Conference on Multimedia Retrieval (ICMR'21), Taipei, Taiwan, 16–19 November 2021.
19. Liu, W.; Duanmu, Z.; Wang, Z. End-to-End Blind Quality Assessment of Compressed Videos Using Deep Neural Networks. In Proceedings of the ACM Multimedia, Amsterdam, The Netherlands, 12–15 June 2018; pp. 546–554.
20. Li, D.; Jiang, T.; Jiang, M. Norm-in-norm loss with faster convergence and better performance for image quality assessment. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 789–797.
21. Li, D.; Jiang, T.; Jiang, M. Unified quality assessment of in-the-wild videos with mixed datasets training. *Int. J. Comput. Vis.* **2021**, *129*, 1238–1257. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.