Xception-based Architecture with Cross-sampled Training for Image Quality Assessment on KonIQ-10k

Tomasz M. Lehmann*, Przemysław Rokita
Warsaw University of Technology, Warsaw, Poland
**Corresponding author: Tomasz M. Lehmann (tomasz.lehmann@dokt.pw.edu.pl)*

Abstract. Image quality assessment is a crucial task in various fields such as digital photography, online content creation, and automated quality control, as it ensures an optimal visual experience and aids in maintaining consistent standards. In this paper, we propose an efficient method for training image quality assessment models on the KonIQ-10k dataset. Our novel approach utilizes a dual-Xception architecture that analyzes both the image content and additional image parameters, outperforming traditional single convolutional models. We introduce cross-sampling methods with random draw sampling of instances from majority classes, effectively enhancing prediction quality in the Mean Opinion Score (MOS) ranges that are underrepresented in the database. This methodology allows us to achieve near state-of-the-art results with limited computing costs and resources. Most importantly, our predictions across the entire spectrum of MOS values maintain consistent quality. Because of using a novel and highly effective method for image sampling, we achieved these results with much lower computational cost, making our approach the most effective way of MOS estimation on the KonIQ-10k database.

Key words: image quality assessment, computer vision, Xception

1. Introduction

Image quality refers to the fidelity of imaging systems in capturing and processing signals to form an image, and the weighted sum of visually important features as perceived by the human eye [1]. This dual perspective is crucial in applications like diagnostics, environmental monitoring, visual media, security, and manufacturing, impacting decision-making and operational efficiency. Both the technical fidelity and subjective appeal highlight the need for robust image quality assessment.

Objective image quality assessment is divided into no-reference, reduced-reference, and full-reference methods, based on the original image's availability. Full-reference metrics compare a test image with the original; reduced-reference uses limited original information, and no-reference assesses the image independently. These methods provide automated metrics for estimating image quality [2].

Blind Image Quality Assessment (BIQA) stands out as the most complex yet most applicable among the three types of image quality assessments because it does not need a reference image. In many cases, such references are not accessible. Deep learning advancements have shown significant potential in enhancing BIQA alongside other areas like image recognition and object identification. The development of BIQA methods would benefit substantially from a vast and varied database that includes naturally occurring image distortions. Nevertheless, the training of deep learning models for

BIQA is currently constrained by the limited scope and synthetic nature of existing databases [3, 4, 5]. Moreover, large-scale quality assessments in a controlled environment are not feasible, given the extensive time and participant involvement required. In the study referenced as [6], the researchers introduced a groundbreaking dataset named KonIQ-10k, consisting of 10,073 images each with an associated quality score. Additionally, they developed a CNN-based model, KonCept512, which surpassed competing models [7,8,9] in performance on both the KonIQ-10k and the LIVE-itW [10] databases.

In this paper, we focused on replicating results with the streamlined Xception architecture [11], which has fewer parameters (22.8 million when the architecture proposed by the original KonIQ-10k dataset authors contains around 56 million variable parameters). Architectures with fewer parameters typically learn faster and are more effective on small datasets because they are less prone to overfitting and require fewer computational resources for training. This is a significant advantage in the context of novel approaches to data distribution during training steps. We explored the effectiveness of a hybrid deep learning model that utilizes dual CNN extractors. Moreover, we have confirmed that incorporating undersampling (due to methods of random sampling from major classes and data duplication in minor classes, referred to as cross-sampling) [13] to reduce training times does not detrimentally impact the overall results. All these additional improvements make our model easier and much faster to train, as well as quicker and lighter for inference.

1.1. Related Works

Image Quality Assessment (IQA) is crucial in many fields, playing a key role in ensuring the precision and efficiency of numerous advanced decision-making and operational systems. IQA is generally divided into subjective and objective categories. Subjective IQA depends on human evaluations, leading to the Mean Opinion Score (MOS) system, which reflects the average perceived image quality. However, its time-consuming and costly nature limits its practical use.

Objective IQA, especially no-reference or Blind Image Quality Assessment (BIQA), has greatly advanced with deep learning. While traditional BIQA methods focused on manually selected features, recent trends lean towards automatic representation learning from raw images to predict quality scores. Deep learning in BIQA, such as the application of deep belief networks by Ghadiyaram et al. and the VGG16 network in DeepBIQ and BLINDER models [14, 15], showcases the effectiveness of deep neural networks in this area. These models estimate image quality by analyzing various image sections and averaging their MOS, considering both individual and overall image quality scores.

Pixel-by-Pixel IQA (pIQA) [16] marks a significant progress in this sector, introducing an innovative way to compute the MOS for each pixel and sum it up for an overall image score. This method surpasses older IQA techniques and aligns closely with human vision, representing a major step forward in objective IQA.

Deep learning-based Full-Reference IQA (FR-IQA) [17] methods have also been developed, focusing on the similarity between an original and altered image. However, their effectiveness varies with the image's complexity, encouraging further exploration of deep learning for more effective feature extraction.

Transfer learning has been utilized to address the challenges of small training datasets in BIQA. Examples include RankIQA [18], which employs a Siamese Network for image quality ranking, and MEON [19], which uses a multi-task approach with shared initial layers for distortion detection and quality prediction. MS-UNIQUE [17], an FR-IQA method, leverages multiple linear decoders trained on large datasets to gauge visual quality by comparing feature vectors of original and distorted images. Talebi et al. proposed a framework based on object-classification architectures [20], while Zhang et al. used a Siamese network for MOS-based image pair ranking [8].

The KonIQ-10k [6] dataset is a notable recent contribution, specifically created for BIQA prediction. The KonCept512 model, based on the Inceptionv2 architecture, has achieved top-tier results on both the KonIQ-10k and LIVE-itW datasets. However, this solution has room for optimization in MOS range prediction and computational efficiency. Inceptionv2, being a large structure, demands significant computing resources, which may pose challenges for training on commonly available and free personal computers and notebooks.

2. Experimental setup

2.1. Dataset

The KonIQ-10k dataset, the largest of its kind for Image Quality Assessment (IQA), includes 10 073 images, each evaluated for quality. Notable for its ecological validity, the dataset prioritizes authenticity in distortion types, content variety, and quality measures. Developed through extensive crowdsourcing, it incorporates over 1.2 million quality evaluations from 1 459 participants, offering a robust foundation for advancing IQA models.

In Figure 1 several images from the specified database are displayed.

For quality indicators, the dataset incorporates measures well-correlated with human perception. These include brightness, colorfulness, Root Mean Square (RMS) contrast, and sharpness, as revealed by preliminary subjective studies. Other factors like image bitrate, resolution, and JPEG compression quality were also assessed. In our research, we narrowed our focus to brightness, contrast, sharpness, and bitrate because these showed a stronger correlation with the Mean Opinion Score (MOS) what is depicted in dataset's correlation matrix (Figure 2). In our study, we employed four key indicators – brightness, contrast, sharpness, and bitrate – as inputs to the Xception architecture [11].

Fig. 1. The image showcases six examples from the KonIQ-10k database, each accompanied by an MOS value derived from the dataset labels positioned either above or below them. Images demonstrating relatively lower quality (MOS < 50) exhibit characteristics such as being cropped, blurred, or noisy. Conversely, as the MOS increases, the images become clearer and more precisely cropped.

This informed the extraction of a feature vector that was integral to our hybrid neural network, allowing for improved image quality predictions based on the KonIQ-10k dataset's findings.

As indicated in the original KonIQ-10k paper, we also divided our dataset into three subsets (training, validation, testing), adhering to the same distribution ratio: 7 058 elements for training, 1 000 for validation, and 2 015 for testing.

2.2. Proposed methods

2.2.1. Model architecture

In our study, we chose the Xception architecture over InceptionResnetv2 [21] due to its efficient use of parameters without compromising on performance. The significantly lower number of trainable parameters was also a crucial factor in our decision, given our intention to use the cross-sampling method, which limits the number of data points in the training set. This choice is supported by comparative evaluations in which Xception surpassed other residual-connected CNNs, including ResNet50, ResNet152 [22], and

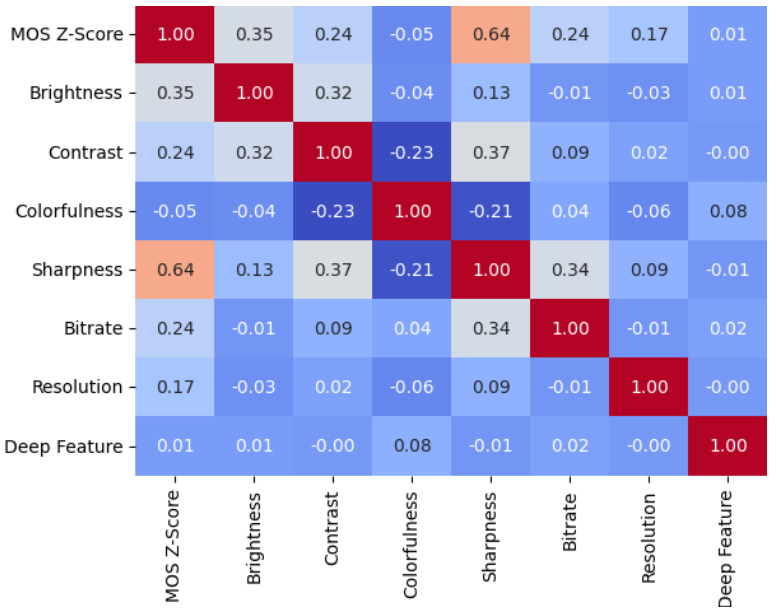

Fig. 2. The pairwise Pearson correlation coefficients visualization among various image quality indicators from the KonIQ-10k dataset. It encapsulates the relationships between Mean Opinion Score (MOS) Z-Score, Brightness, Contrast, Colorfulness, Sharpness, Bitrate, and Resolution, along with extracted Deep Features. The depicted correlations offer a concise overview of the interdependencies between perceived image attributes.

SENet154 [23], in terms of processing efficiency while still maintaining competitive accuracy. This efficacy positions Xception as the preferred model for our image analysis tasks. The most significant differences between the Xception and InceptionResNetV2 architectures lie in their structural design and efficiency.

Xception revolutionizes the traditional Inception architecture by adopting depthwise separable convolutions, which streamline the model by reducing the number of parameters without sacrificing efficiency. Unlike InceptionResNetV2, which enhances the Inception model with residual connections for increased depth and complexity, potentially improving accuracy at the expense of greater computational demands, Xception optimizes for both computational efficiency and performance. This is achieved by decoupling the mapping of cross-channel and spatial correlations in the feature maps, a strategy that allows Xception to surpass the performance of its Inception counterparts in benchmark tasks while requiring fewer computational resources.

Xception is an advanced deep learning model that utilizes depthwise separable convolutions as a fundamental building block, optimizing computational efficiency and model

performance. It diverges from InceptionResNetV2 by employing depthwise separable convolutions, which decouple the mapping of cross-channel correlations and spatial correlations in feature maps, instead of the Inception modules with mixed convolutions. This architectural choice facilitates a more efficient use of model parameters and enables the Xception network to outperform its Inception counterparts on benchmark tasks with fewer computational resources. In our research, we observed that a single training step for the Xception model is around twice as fast.

The Xception framework, embodying "Extreme Inception", is composed of 36 convolutional layers structured into 14 modules, all based on depthwise separable convolutions (Fig. 3). This design principle posits that the correlations within the feature maps of convolutional neural networks can be effectively separated, leading to a model that is both powerful and efficient. The architecture, characterized by its simplicity akin to the VGG16 model but diverging from the more intricate Inception designs, is detailed in its foundational publication [11]. Our findings indicate that Xception's training process is notably faster, with a single step taking roughly half the time compared to more complex models.

To assess the impact of extracted image parameters (brightness, contrast, sharpness, and bitrate) on the quality of results, we utilized a pioneering dual-Xception architecture. One Xception model, pre-trained on the ImageNet dataset [24], was used to extract a feature tensor the size of the original architecture's last linear layer (1024 neurons). A second Xception was adapted to accept four floating-point values (the aforementioned parameters) as input and was not pre-trained. The tensor returned by this part of the proposed structure also had a dimension of 1024. Outputs from both models were then merged using a matrix concatenation operation. The combined feature vector (of size 2048) was processed through a Leaky ReLU activation function and a final linear layer with a single output size.

The Dual-Xception architecture is presented in the diagram in Figure 4.

While this approach increased computational complexity, the goal was to evaluate the model's sensitivity to the parameters introduced into the second model. We examined the models' performance across six scenarios: image-only Xception (with and without cross-sampling), parameters-only Xception (with and without cross-sampling), and the combined architecture (with and without cross-sampling).

In the initial stage, we also compared the proposed dual-Xception architecture with more traditional methods. Our goal was to develop a single, cohesive structure capable of learning from both image data and additional parameter information. While it was possible to use two or three separate models for this purpose, maintaining the entire training pipeline in a unified form was crucial, where we trained just one neural network function using only one optimizer and loss function.

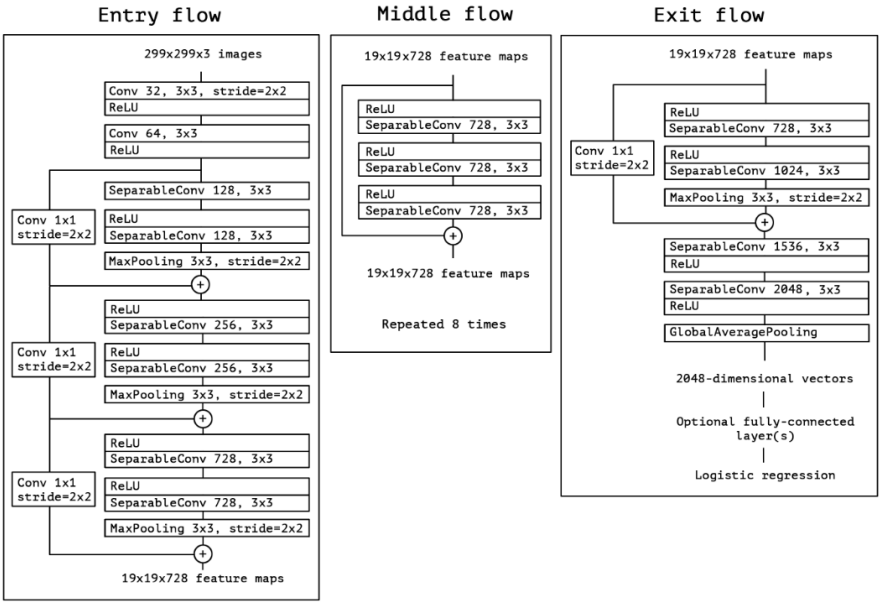

Fig. 3. The complex structure of the Xception architecture is derived directly from the paper [11] (replicated here from [12] according to arXiv License Information). It is important to note that all convolutions are followed by batch normalization, and all SeparableConvolution structures use a depth multiplier.

The dual-Xception model's sophisticated approach offers distinct advantages by separately processing scalar attributes with an Xception model, rather than merely concatenating them with image features or using a basic linear layer. This method acknowledges the complexity of scalar attributes like brightness and contrast, allowing for a nuanced abstraction and integration with image-derived features. It enhances the model's ability to grasp the intricate, non-linear relationships between these attributes and image quality, potentially improving accuracy.

However, the dual-Xception framework's complexity and computational demands are notable drawbacks, increasing training time and data requirements to avoid overfitting. The complexity may also complicate model adjustments and necessitate meticulous regularization.

We decided to evaluate the proposed novel architecture by comparing the mentioned architecture with simpler networks illustrated in the Figure 5.

On the left, we observe that a simple linear neural network layer is employed as a replacement for the parameter-focused Xception part, thereby reducing computational

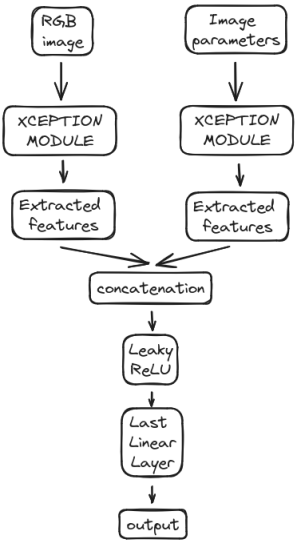

Fig. 4. The Dual-Xception architecture consists of two primary components. The first is the standard Xception model trained on RGB images. The second is a modified version of Xception, designed to process a 4-dimensional input reflecting image parameters: brightness, contrast, sharpness, and bitrate. The duality lies in how these separate models are combined: features extracted from both networks are concatenated and then passed through additional network layers (including an activation function and a linear layer) to predict the final image quality, measured by the Mean Opinion Score (MOS).

complexity. The resulting tensor was then concatenated with features extracted from the RGB image. On the right, we note that scalars are treated as tensors themselves without any preprocessing or feature engineering methods.

2.2.2. Loss function

In this paper, we propose the use of Mean Square Error (MSE) as a metric for assessing the average squared difference between the estimated Mean Opinion Score (MOS) and the labeled ground truth. MSE is a widely adopted approach in a multitude of computer vision-based predictive models. The equation for the loss function is presented below:

$$\mathrm{MSE}(x, y) = \frac{1}{n} \sum_{n} (x_n - y_n)^2 \,, \tag{1}$$

where:
n – number of images in the batch,
x_n – ground truth MOS,

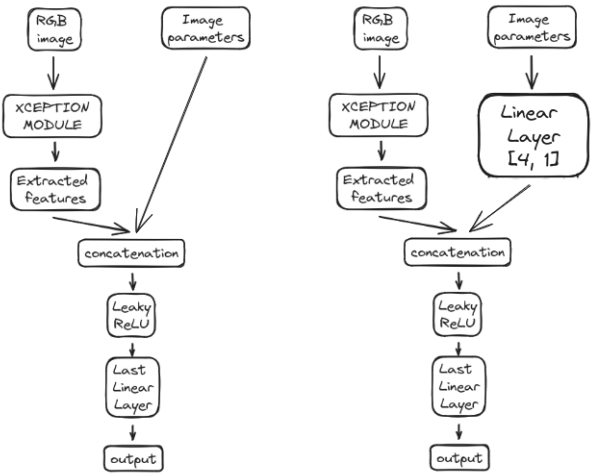

Fig. 5. On the left, we see an architecture where a linear layer replaces the parameter-oriented Xception module. The input size matches the number of included parameters, while the output size is set to 1. On the right, we treat the scalar values of parameters as features in themselves, without proposing any transformations except tensorification. In both cases, a crucial step is the concatenation operation, where features extracted from the RGB image are merged with information extracted from parameters in two distinct manners.

y_n – predicted MOS.

To assess the impact of cross-sampling on MSE within narrower MOS intervals, we tracked this metric across the following ranges: 0-20, 20-40, 40-60, 60-80, and 80-100. The data was categorized according to the labeled MOS values.

2.2.3. Cross-sampling

To optimize computing time and enhance results in MOS ranges with fewer training examples, we employed an cross-sampling strategy.

Under-sampling is a technique employed to address imbalances in datasets by decreasing the size of the more dominant class to match that of the minority class. This method is part of a suite of tools that data scientists use to extract more accurate insights from datasets that initially exhibit a skew in class distribution. In our solution, we have also incorporated randomness into the drawing of samples at every step. Therefore, we prefer to refer to this procedure as cross-sampling, and we adhere to this terminology throughout this paper.

Given the substantial imbalance in our dataset across the ranges 0-20, 20-40, 40-60, 60-80, and 80-100, we applied a straightforward algorithm to ensure the model paid greater attention to underrepresented classes. If the number of images in a range was

more than double but less than quadruple the quantity in the smallest range, the dataset was randomly reduced during each training epoch to a maximum of twice the size of the smallest class. If the quantity was between four to six times larger, the limit was set to three times the size of the smallest class, and so forth, capping at four times the size for any larger quantities. This approach effectively created a more balanced training environment, promoting better learning from minority classes.

2.2.4. Training procedures

The model training was conducted on an NVidia RTX 3070 GPU with 8GB memory. Each training session involved 60 epochs, with a batch size of 4, at a resolution of 512x384. The duration of training varied between 5 to 20 hours, depending on the architecture and whether the dataset was processed with cross-sampling. The number of training epochs was suggested by the authors of KonIQ-10k, and in the subsequent chapters, we demonstrate that it was not a very accurate approximation. After each epoch, the model was evaluated on a validation set to monitor for signs of overfitting. Ultimately, the best-performing model – characterized by the lowest loss – was selected for testing. The ADAM algorithm [25], known for its adaptive learning rate methods, served as the optimizer. The initial learning rate was set at 0.0001, halving every 20 epochs.

For our metrics, we utilized PLCC and SROCC, both of which are widely employed in the evaluation of image quality.

PLCC stands for Pearson Linear Correlation Coefficient, which measures the linear correlation between two variables, providing a value between -1 and 1. A PLCC of 1 indicates perfect positive correlation, while -1 indicates a perfect negative correlation. It is often used in image quality assessment to compare the similarity between the quality ratings of images by an algorithm and subjective ratings by humans.

SROCC denotes Spearman's Rank Order Correlation Coefficient, a non-parametric measure of rank correlation. It assesses how well the relationship between two variables can be described using a monotonic function. In the context of image quality assessment, it ranks the images based on quality and compares the algorithm's rankings with those from human assessments.

$$\mathrm{PLCC}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2 \sum_{i=1}^{n}(Y_i - \overline{Y})^2}}\,, \tag{2}$$

where:
n – number of observations,
X_i – i^{th} observation of variable X,
Y_i – i^{th} observation of variable Y,
$\overline{X}$ – mean of all observations of variable X,
$\overline{Y}$ – mean of all observations of variable Y.

$$\mathrm{SROCC}(X, Y) = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}, \tag{3}$$

where:
n – number of observations,
d_i – difference between the ranks of the i^{th} observations of variables X and Y.

3. Experimental results

The authors of the referenced paper conducted experiments to find the optimal input resolution for their model by training on the original resolution (1024×768) and two lower resolutions (512×384 and 224×224). They found that the models trained on the smallest resolution (224×224) performed worse than the others, suggesting a significant loss of quality-related information during the down-sampling process. Interestingly, the models trained on the medium resolution (512×384) outperformed those trained at the original resolution.

For our study, we exclusively utilized the Xception architecture, pre-trained on the ImageNet dataset. The results demonstrated that both PLCC and SROCC metrics were slightly better for the 512×384 resolution compared to the 1024×768, and substantially better than the 224×224 resolution.

These findings led us to exclusively utilize the 512×384 resolution in subsequent operations.

3.1. Comparative Analysis of Dual-Xception Architectures

In this analysis, we evaluate the effects of modifications to the Xception architecture on performance outcomes. Initially, the original Xception model was trained using the KonIQ-10k image dataset.

Subsequently, we compared three models derived from architectures proposed in section 2.2.1.

The first and simplest architecture incorporates tensored parameters as additional features, combining this information with features extracted from the RGB image through concatenation.

The second approach introduces a simple linear layer to process parameters, functioning as a parameter-oriented branch of the architecture.

The third solution employs a modified Xception model designed to accept a 4-dimensional input of image parameters: brightness, contrast, sharpness, and bitrate, in lieu of standard RGB images. In this design, the parameters are fed into the initial layer of the modified Xception network, effectively substituting the first convolutional layer that typically processes RGB values. While a Multi-Layer Perceptron (MLP) could handle these

Tab. 1. Results comparing three methods of integrating features from images with image parameters are presented. The first method, labeled 'Plain Parameters', involved no preprocessing but directly combined tensored parameter values. The second method, 'Linear Layer', utilized a linear neural network to extract features from image parameters. The third method utilized our novel dual-Xception architecture.

	Plain parameters	**Linear Layer**	**dual-Xception**
PLCC	0.912	0.916	0.920
SROCC	0.896	0.898	0.903
MSE	6.32	6.22	6.06
MSE (0-20)	10.18	13.31	12.01
MSE (20-40)	8.82	8.41	8.48
MSE (40-60)	6.73	6.60	6.66
MSE (60-80)	5.10	4.87	4.56
MSE (80-100)	6.61	7.24	6.10
Parameters	20.8M	20.8M	41.6M

parameters, the modified Xception model still utilizes depthwise separable convolutions, which are central to Xception's design. This approach might leverage the convolutions for effective feature extraction and transformation from non-image data, representing a novel strategy that could treat the spatial hierarchies and patterns within the parameters similarly to image features. Future research could explore different architectures to refine this extractor and further reduce computational demands.

We evaluated the models using three main metrics: PLCC, SROCC, and MSE, as detailed in the previous chapter. Additionally, we assessed the MSE within the ground truth range of the MOS parameters, dividing these ranges into five categories (MOS from 0 to 20, MOS between 20 and 40, etc.). A lower MSE value indicates more accurate predictions. For PLCC and SROCC, the ideal value is 1, with the value decreasing as prediction quality deteriorates (down to a minimum value of 0).

In Table 1 we present results comparing three methods of combining features extracted from images with image parameters. In the first column, titled "Plain Parameters", we show the metrics for the method where we used no preprocessing but combined tensored parameter values directly. In the middle column, titled "Linear Layer", we present results achieved by using a linear neural network as a feature extractor from image parameters. In the third column, we present results for our pioneering dual-Xception architecture. We observe that the differences in results are relatively small. However, in most cases, the dual-Xception-based network achieved better results.

The convolutional-based extractor might be better for scalars because it can capture non-linear relationships and subtle patterns within the data, which a simple linear model might overlook. Given the high performance of current models, even slight improvements are considered significant achievements in the field. This underscores the potential of

Tab. 2. Training results using standard Xception on KonIQ-10k images (Images), Xception trained on tabular data comprising image parameters (Parameters), and the outcomes using the Dual-Xception a approach (Images+Parameters).

	Images	**Parameters**	**Images+Parameters**
PLCC	0.91	0.74	0.92
SROCC	0.90	0.74	0.90
MSE	6.32	10.34	6.06
MSE (0-20)	11.17	26.42	12.01
MSE (20-40)	8.18	14.44	8.48
MSE (40-60)	6.64	9.28	6.66
MSE (60-80)	5.38	8.63	4.56
MSE (80-100)	5.91	15.08	6.10

convolutional approaches, like the dual-Xception architecture, in enhancing predictive accuracy, even in scenarios where traditional models already perform exceptionally well.

In Table 2 we analyzed three constructed models: the standard Xception architecture (termed "Images"), Xception trained solely on the tabular image attributes parameters such as brightness, contrast, sharpness, and bitrate (referred to as "Parameters"), and a combined architecture. This combined model integrates feature extractors from the first two models and produces a corrected result through a linear neural layer, employing the concatenation of tensors from the initial models.

The table shows that the Dual-Xception model excelled, particularly in assessing image quality within the 60 to 80 MOS range. This superior performance is likely due to the specific parameters within this range being highly indicative of image quality, as demonstrated by the improved results in the same quality range when using only the "Parameters" model. Notably, we did not employ the cross-sampling technique in this study, which might lead to skewed results from an imbalanced dataset.

In the graphs presented in Fig. 6 we observe the variation in the SROCC and PLCC metrics throughout the training epochs. It is evident that the Dual-Xception model initially registered lower values compared to the conventional approach. However, around the 8th to 10th epoch, the performance of the standard approach plateaued, whereas the more complex Dual-Xception architecture continued to improve as it progressed further in training.

3.2. Comparative Analysis of Models with Cross-Sampling

The images in the KonIQ-10k dataset are significantly unbalanced, impacting the final results and making model training less effective. This is demonstrated in Table 3, where the number of samples in the smallest and largest groups differs by nearly 30 times.

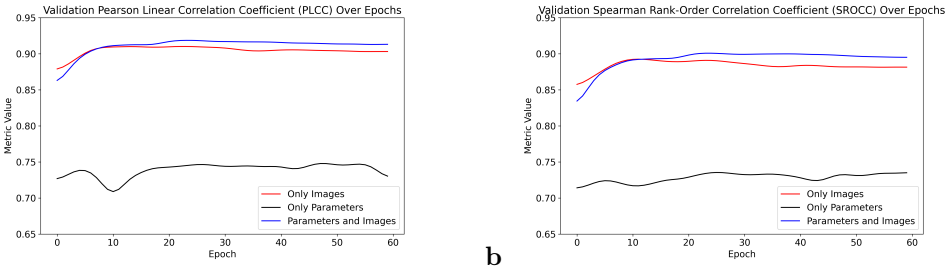

Fig. 6. Validating with two measures in function of epochs for the three types of models: (**a**) PLCC; (**b**) SROCC. Function color represents applied architecture (blue - Dual-Xception, red - Image Xception, black - Parameters Xception

To enhance the predictability of our models for lower-quality ranges, we employed and compared several cross-sampling methods. Additionally, we experimented with weighted MSE loss to focus the algorithm more on better predictions in the minority classes. In table below, we can observe the exact quantity of images in every range of MOS in the KonIQ-10k dataset. We note that in the class where MOS is between 0 and 20, there are fewer than 200 image instances, while the sum of images with MOS between 60 and 80 is 30 times larger.

In the preliminary phase of our research on under and over-sampling, we explored three distinct methodologies. The initial method involved constraining the number of elements in each class to correspond with the highest count found in the smallest class for the training dataset (method 1). Here, *class* denotes a grouping of instances within identical Mean Opinion Score (MOS) ranges. For every training iteration, instances were randomly selected from the categorized dataset, ensuring varied images for classes with a higher image count. In the second method, we limited the number of elements to the minimum count unless the image count exceeded 1 000. In such cases, we increased the image count for the specified range by double (method 2), leading to a twofold increase in data for the 40-60 and 60-80 ranges compared to other classes. The third approach

Tab. 3. Number of images samples in each of the five MOS ranges.

MOS range	**Samples**
(0, 20>	183
(20, 40>	1189
(40, 60>	3081
(60, 80>	5408
(8, 100>	212

Tab. 4. Comparison of the outcomes from four data sampling techniques.

	Without C-S	**Method 1**	**Method 2**	**Method 3**
PLCC	0.91±0.01	0.89±0.01	0.91±0.01	0.91±0.02
SROCC	0.88±0.01	0.87±0.01	0.89±0.01	0.89±0.02
MSE	6.40±0.09	7.30±0.12	6.30±0.15	7.35±0.20
MSE (0-20)	8.64±0.11	10.01±0.13	9.46±0.15	7.99±0.31
MSE (20-40)	8.61±0.11	8.57±0.16	8.41±0.14	8.14±0.20
MSE (40-60)	6.83±0.08	8.12±0.20	6.78±0.18	7.77±0.22
MSE (60-80)	5.42±0.05	6.37±0.24	5.24±0.18	6.89±0.25
MSE (80-100)	5.54±0.08	5.31±0.06	5.64±0.08	5.12±0.10
Ranges standard deviation	1.57±0.15	1.85±0.21	1.80±0.18	0.72±0.27

(method 3) employed a similar technique, but the number of elements in the largest classes was quadrupled relative to the smallest class. For three other ranges, we tripled the number of image instances, necessitating the repeated use of identical images. To mitigate overtraining risks, we applied image augmentation techniques, including rotations, flips, minor contrast modifications, and noise addition, while avoiding substantial alterations to maintain the integrity of the MOS ground truth.

During the validation phase, we modified our criteria for selecting the optimal model. Previously, the model with the lowest total Mean Squared Error (MSE) on the validation set was chosen. However, given our emphasis on minimizing the standard deviation across all five MOS ranges in this phase, we adopted a different approach. The optimal model was now determined based on the minimal sum of MSEs from each of the five MOS ranges.

It is important to highlight that the randomness in data allocation and manipulation, especially for the largest class, led to a relatively high standard deviation, thereby compromising the model's predictability. This study entailed ten training iterations per method, with notable disparities in results for edge cases.

The results for each model are presented in the table below. To ensure a more reliable comparison, we employed the same model selection criteria for validation across every method, including the primary architecture discussed in the previous chapter. This is why the results do not completely align with those presented in Tab. 1.

As depicted in the table above, due to the substantial reduction of data, cross-sampling can significantly decrease computing time without compromising performance quality. The data in the KonIQ-10k database are not evenly balanced, presenting a valuable opportunity to optimize our models even with relatively modest computing resources.

What is important to mention is that due to limitations of the dataset in every single training step, the time of the training procedure was reduced four times for method

Tab. 5. The table presents the computing time for a single training epoch for four selected models. From the left, we have the KonCept512 model provided by the KonIQ-10k authors, our dual-Xception model without cross-sampling implementation, and three cross-sampling methods explained above in this chapter.

	KonCept512	**Without C-S**	**Method 1**	**Method 2**	**Method 3**
Time	273 s	357 s	54 s	85 s	110 s
Parameters	56.0 M	40.8 M	40.8 M	40.8 M	40.8 M

number 3 and up to 7 times for the most limited method 1. This makes it a significantly more effective method, capable of enhancing computations by several times. It is a direct result of the dataset sampling procedure where we operate only on a small subset of the original dataset.

In Table 5 we illustrate the changes in computational time per epoch following the implementation of a given cross-sampling technique and compare this to the computing time of the current state of the art. The presented time refers to the duration of a single training epoch iteration. Similar to the authors of the state-of-the-art method, we trained our models over 60 epochs. The given time represents the average value from the entire training process. The training was conducted on an NVIDIA RTX 3070 GPU with 8GB of memory and a batch size of 4.

We can observe that the computational time for the original KonCept512 model is shorter than that of our dual-Xception architecture without the implementation of the cross-sampling method. This discrepancy may be due to different implementation approaches. The authors do not provide a PyTorch implementation of their network, and for this analysis, we opted to use the HuggingFace implementation, which might be slightly more optimized. However, we can still see that by employing our training procedure, which depends on the dataset's quantitative operations, we were able to outperform this state-of-the-art model by more than fivefold.

3.3. Comparison with other algorithms

We compared our results from the normal training procedure with our outcomes where we used a drastically limited dataset, following cross-sampling methods, against other solutions that were or still are seen as state-of-the-art in blind image assessment. Results are presented in Tab. 6.

We observed that the KonCept512 model remains the most effective. However, none of the authors of the papers included in our comparison provided details on accuracy changes across the entire spectrum of the MOS parameter. We have developed a very simple, easy-to-train, and extremely fast solution that guarantees prediction quality for all ranges of MOS values.

Tab. 6. Comparison of performance scores of several well-known and influential methods on the KonIQ-10k dataset. Results for each model, except our own one, are derived from [6].

Method	**SROCC**	**PLCC**
BIQI [26]	0.56	0.62
BLIINDS-II [27]	0.59	0.60
BRISQUE [28]	0.71	0.71
CNN [4]	0.57	0.59
DeepBIQ (VGG16) [5]	0.87	0.89
DeepBIQ (InceptionResNetV2) [5]	0.91	0.91
KonCept512 [6]	0.92	0.94
Ours without C-S	0.90	0.92
Ours with C-S	0.89	0.91

4. Conclusions

We introduced an efficient yet easy-to-train model that achieved near state-of-the-art performance with a dataset significantly smaller in size. Our method ensures excellent predictions across the entire MOS parameter range on the KonIQ-10k dataset. By using the cross-sampling method, we optimized single epoch processing time by up to 5 times without a significant decrease in result quality. We can assert that our solution is easier and faster for training and inference. The achieved results are also examined across the entire spectrum of MOS image values, making our model unique.

For future work, we could delve into more sophisticated research regarding optimal under/over-sampling techniques. Our proposed methodology, while experimental, suggests that different dataset partitioning might yield even better results. Another aspect worth investigating is the application of weighted training loss that varies according to the ground truth parameter value. However, this approach may not significantly reduce training time, especially if we continue to utilize sampling methods.

We also propose to intensify research on how to optimize feature extractors that operate on numerical data. We need to investigate alternatives to the Xception architecture that may offer comparable results.

References

[1] N. Burningham, Z. Pizlo, and J. P. Allebach. Image Quality Metrics. In Hornak, Joseph P. (ed.). *Encyclopedia of imaging science and technology*, Wiley, New York, 2002. doi:10.1002/0471443395.img038.

[2] I. H. AL-Qinani. A Review Paper on Image Quality Assessment Techniques. *International Journal of Modern Trends in Engineering & Research*, 6(8):1–7, 2019. doi:10.21884/IJMTER.2019.6023.SVDQQ.

[3] S. Bosse, D. Maniry, T. Wiegand, and W. Samek. A deep neural network for image quality assessment. *Proc. IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3773–3777. doi:10.1109/ICIP.2016.7533065.

[4] L. Kang, P. Ye, Y. Li, and D. Doermann, Convolutional neural networks for no-reference image quality assessment. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1733–1740, 2014. doi:10.1109/CVPR.2014.224.

[5] S. Bianco, L. Celona, P. Napoletano, and R. Schettini. On the use of deep learning for blind image quality assessment. *Signal, Image and Video Processing*, 12(2)355–362, 2018. doi:10.1007/s11760-017-1166-8.

[6] V. Hosu, H. Lin, T. Sziranyi, and D. Saupe. KonIQ-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE Transactions on Image Processing*, 29:4041–40561, 2020. doi:10.1109/tip.2020.2967829.

[7] V. R. Dendi, C. Dev, N. Kothari, and S. S. Channappayya. Generating image distortion maps using convolutional autoencoders with application to no reference image quality assessment. *IEEE Signal Processing Letters*, 26(1):89–93, 2018. doi::10.1109/LSP.2018.2879518.

[8] W. Zhang, K. Ma, G. Zhai and and X. Yang. Learning to blindly assess image quality in the laboratory and wild. *Proc. 2020 IEEE International Conference on Image Processing (ICIP)*, pp. 111–115, 2020. doi:10.1109/ICIP40778.2020.9191278.

[9] D. Varga, T. Szirányi, and D. Saupe. DeepRN: A content preserving deep architecture for blind image quality assessment. *Proc. 2018 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2018. doi:10.1109/ICME.2018.8486528.

[10] D. Ghadiyaram and A. C. Bovik. Massive online crowdsourced study of subjective and objective picture quality. *IEEE Transactions on Image Processing*, 25(1):372–387, 2016. doi:10.1109/TIP.2015.2500021.

[11] F. Chollet, Xception: Deep learning with depthwise separable convolutions. *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, 2017. doi:10.1109/CVPR.2017.195.

[12] F. Chollet, Xception: Deep learning with depthwise separable convolutions. *arXiv*, preprint arXiv:1610.02357, 2017. doi:10.48550/arXiv.1610.02357.

[13] R. Mohammed, J. Rawashdeh and M. Abdullah. Machine learning with oversampling and undersampling techniques: Overview study and experimental results. *Proc. 2020 11th International Conference on Information and Communication Systems (ICICS)*, pp. 243–248, 2020. doi:10.1109/ICICS49469.2020.239556.

[14] D. Ghadiyaram and A. C. Bovik. Massive online crowdsourced study of subjective and objective picture quality. *IEEE Transactions on Image Processing*, 25(1)372–387, 2016. doi:10.1109/TIP.2015.2500021.

[15] F. Gao, J. Yu, S. Zhu, Q. Huang, and Q. Tian. Blind image quality prediction by exploiting multi-level deep representations. *Pattern Recognition*, 81:432–442, 2018. doi:10.1016/j.patcog.2018.04.016.

[16] W.-H. Kim, et al. Pixel-by-pixel Mean Opinion Score (pMOS) for no-reference image quality assessment. *ArXiv*, preprint arXiv:2206.06541, 2022. doi:10.48550/arXiv.2206.06541.

[17] M. Prabhushankar, D. Temel, and G. AlRegib. MS-UNIQUE: Multi-model and sharpness-weighted unsupervised image quality estimation. *Electronic Imaging*, 29(12):30–35:art00006, 2017. doi:10.2352/ISSN.2470-1173.2017.12.IQSP-223.

[18] X. Liu, J. van de Weijer, and A. D. Bagdanov. RankIQA: Learning from rankings for no-reference image quality assessment. *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 1040–1049, 2017. doi:10.1109/ICCV.2017.118.

[19] K. Ma, W. Liu, K. Zhang, Z. Duanmu, Z. Wang, and W. Zuo. End-to-end blind image quality assessment using deep neural networks. *IEEE Transactions on Image Processing*, 27(3):1202–1213, 2018. doi:10.1109/TIP.2017.2774045.

[20] H. Talebi and P. Milanfar, NIMA: Neural image assessment. *IEEE Transactions on Image Processing*, 27(8):3998–4011, 2018. doi:10.1109/TIP.2018.2831899.

[21] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. *Proc. 31st AAAI Conference on Artificial Intelligence*, Vol. 31, No. 1, pp. 4278–4284, 2017. doi:10.1609/aaai.v31i1.11231.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi:10.1109/CVPR.2016.90.

[23] J. Hu, L. Shen, and G. Sun. Squeeze-and-Excitation Networks. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018. doi:10.1109/CVPR.2018.00745.

[24] J. Deng, W. Dong, Socher, R., Li-Jia Li, Kai Li, Li Fei-Fei. ImageNet: A large-scale hierarchical image database. *Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi:10.1109/cvprw.2009.5206848.

[25] D. P. Kingma, J. Ba. Adam: A method for stochastic optimization. *Proc. 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 7–9, 2015. doi:https://arxiv.org/abs/1412.6980.

[26] A. Moorthy and A. Bovik. A two-step framework for constructing blind image quality indices. *IEEE Signal Processing Letters*, 17(5):513–516, 2010. doi:10.1109/LSP.2010.2043888.

[27] M. A. Saad, A. C. Bovik, and C. Charrier. Blind image quality assessment: A natural scene statistics approach in the DCT domain. *IEEE Transactions on Image Processing*, 21(8):3339–3352, 2012. doi:10.1109/TIP.2012.2191563.

[28] A. Mittal, A. K. Moorthy, and A. C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695-4708, 2012. doi:10.1109/TIP.2012.2214050.