An Ensemble of ConvNeXt Models for Perceptual Quality Assessment of Filtered Images

Sergio Sanz-Rodríguez
Berlin, Germany
sergio.sanz.rodriguez@gmail.com

***Abstract*— The widespread use of aesthetic filters on social media introduces new challenges for Image Quality Assessment (IQA), as traditional distortion-based metrics often fail to capture the subjective and content-aware characteristics of these enhancements. This paper proposes a no-reference IQA model based on the ConvNeXt-Large architecture, which incorporates tailored training strategies, along with careful data pre-processing and post-processing techniques to boost predictive accuracy. An ensemble of models further enhances the robustness and generalization of the predictions across diverse filter types. Experimental results demonstrate that the proposed method surpasses the baseline in terms of PLCC (0.690 vs. 0.543), SROCC (0.654 vs. 0.516), and RMSE (0.104 vs. 0.121), confirming its effectiveness in assessing the perceptual quality of filtered images.**

Keywords—ConvNeXt, deep learning, ensemble learning, image quality assessment, perceptual quality, filtered images

I. Introduction

With the growing popularity of social media, aesthetic filters have become a common tool for modifying images, widely used to stylize and enhance shared visual content. Unlike classical image degradations, such as compression artifacts, noise, or blurring, these filters introduce subjective, content-sensitive changes that prioritize artistic enhancement over fidelity. This distinction leads to unique challenges for computational Image Quality Assessment (IQA) methods, which traditionally focus on quantifying objective distortions.

Commonly used IQA metrics, including the Video Multimethod Assessment Fusion (VMAF) [1] and Structural Similarity Index (SSIM) along with its variants [2], are designed to measure traditional degradation types and thus struggle to reliably evaluate the perceptual quality of images altered by aesthetic filters. To address this gap, Wu et al. [3] introduced a state-of-the-art, no-reference IQA method specifically targeting filter-enhanced images. Their model leverages a dual ConvNeXt backbone [4] that extracts two complementary feature representations: one focusing on distortion-related features, while the other targets filter-aware features. These features are subsequently fused using a combination of local and global contextual attention mechanisms, culminating in a regression head that pools fused information to predict a Mean Opinion Score (MOS) that correlates well with human perception.

Building on this foundation, this paper presents a no-reference IQA model designed for the VCIP 2025 competition on Image Manipulation Quality Assessment Challenge [5]. The proposed approach utilizes an ensemble of ConvNeXt networks to enhance generalization capability and incorporates specialized training techniques and processing strategies aimed at improving predictive accuracy for filter-manipulated images. We model is evaluated using the official challenge dataset and benchmarked against the baseline method [3], whose reference implementation is publicly available in [5].

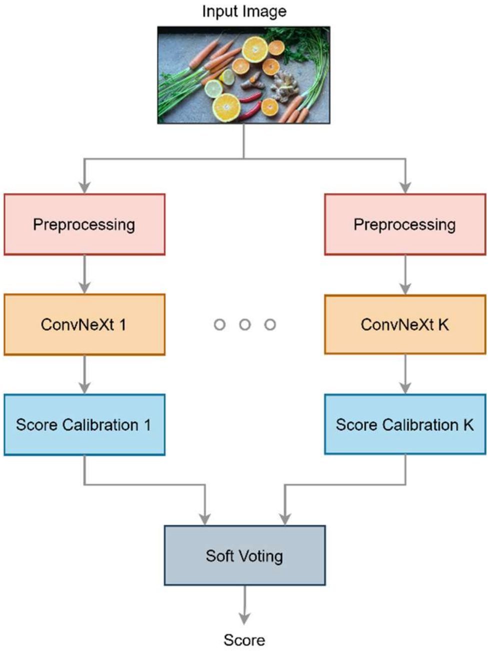

Fig. 1. The proposed no-reference, multi-model IQA pipeline

The remainder of this paper is organized as follows. Section II provides an overview of the proposed IQA model. Sections III through VI detail the key components of the processing pipeline, including the ensemble strategy as well model training and cross-validation procedures. Section VII describes the experimental setup, evaluation metrics, and performance comparisons. Finally, Section VIII draws conclusions and outlines potential directions for future work.

II. General Overview

The proposed IQA pipeline is illustrated in Fig. 1. A total of K ConvNeXt models process the input image and generate MOS scores ranging from 0 to 1. In the experiments, K is set to 8. Before making predictions, the image is preprocessed to match the input format expected by the models. After each ConvNeXt model makes its prediction, the score is recalibrated using a sigmoid function to improve predictive accuracy. Finally, the predictions are ensembled using soft voting by computing their arithmetic mean.

III. Image Preprocessing

Two types of preprocessing strategies are applied to the input image: 1) general preprocessing and 2) training preprocessing.

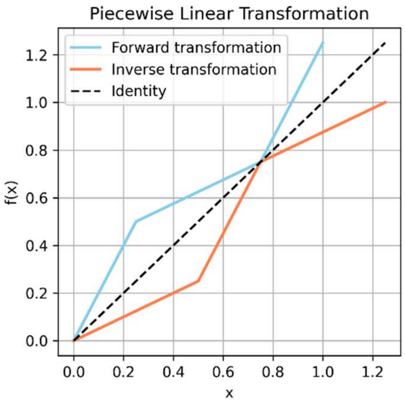

Fig. 2. Piecewise linear transformation applied to the MOS scores

A. *General Preprocessing*

Given the high resolution of the training images (1920×1080 pixels), they are initially downscaled to 910×512 pixels to decrease computational requirements and better fit available CPU and GPU resources. Afterwards, the image is normalized using the standard ImageNet dataset statistics, with mean values of [0.485, 0.456, 0.406] and standard deviations of [0.229, 0.224, 0.225].

B. *Training Preprocessing*

To increase diversity in the training dataset and improve generalization, several augmentation techniques are applied:

- *Data augmentation:* This step involves horizontal flipping, cropping, and affine transformations (rotation and translation). Other common techniques, such as color jittering and blurring, were intentionally excluded to preserve the original artistic intent of the image filters.
- *MOS transformation:* The distribution of scores in the training dataset is overrepresented in the middle range of the MOS scale, which may result in reduced accuracy at the lower and higher ends. To address this imbalance, a piecewise linear transformation is applied to the MOS labels (see Fig. 2). This operation stretches the low and high ends of the score range while compressing the middle region. In this way, the model will prioritize learning in perceptually critical quality extremes. At inference time, the inverse transformation is applied to recover the original scale. This approach preserves the original data distribution and avoids introducing sampling bias.

Note that all these training augmentation techniques are **optional**, but they may improve model generalization.

IV. The ConvNeXt Architecture

A wide range of deep learning models is available in the literature, many with existing PyTorch implementations [6]. For this computer vision challenge, several state-of-the-art architectures were evaluated, including Swin Transformer, EfficientNet, and ConvNeXt. After a round of pre-training, the ConvNeXt-Large model was found to offer the best trade-off between predictive accuracy and computational cost. Notably, this architecture is the same as that used in the provided baseline.

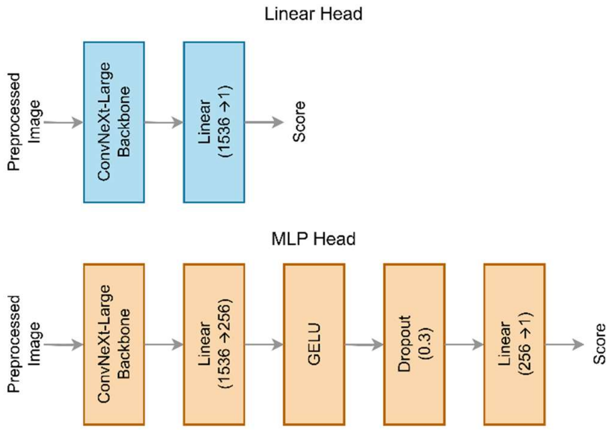

Fig. 3. ConvNeXt-Large backbone with a Linear Head (top) and an MLP Head (bottom)

A. *Backbone*

ConvNeXt is a modern Convolutional Neural Network (CNN) that integrates design principles inspired by Vision Transformers (ViT) [7] while preserving the computational efficiency of traditional CNN. The Large variant comprises 196.2 million parameters.

B. *Regression Heads*

Two regression heads are explored in this work and illustrated in Fig. 3: 1) a Linear Head and 2) a Multi-Layer Perceptron (MLP) Head:

- The *Linear Head* consists of a single fully connected layer that maps the 1536-dimensional output of the backbone to the final MOS score.
- The *MLP Head* includes a linear layer (1536 → 256), followed by a GELU activation, a dropout layer with a 30% probability, and a final linear layer to produce the MOS prediction.

V. Loss Functions

To train a neural network, backpropagation is a widely used method for computing parameter updates. Specifically, it calculates the gradient of a loss function with respect of the weights of the network.

In this work, three loss function algorithms are explored: Mean Squared Error (MSE), Huber Loss, and Weighted MSE.

A. *MSE Loss*

MSE is the most common loss function used in regression tasks and is defined as:

$$MSE = \frac{1}{L}\sum_{i=0}^{L-1}(y-\tilde{y})^2 \tag{1}$$

where y and $\tilde{y}$ represent the ground-truth and predicted values, respectively, and L denotes the number of samples.

B. *Weighted MSE Loss*

Due to the underrepresentation of the low and high MOS regions in the training dataset, a larger weighing factor is applied to those regions in the MSE formulation, as shown in Eq. (2):

$$WMSE = \frac{1}{L}\sum_{i=0}^{L-1}\omega(y-\tilde{y})^2 \tag{2}$$

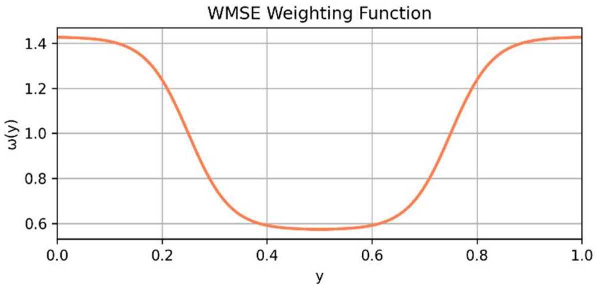

Fig. 4. Weighting function for the WMSE Loss

The weighing function, denoted as $\omega(y)$, is illustrated in Fig. 4, where two symmetric sigmoid curves centered around the midpoint (0.5) are combined to emphasize the extremes of the MOS distribution.

A. Huber Loss

Huber Loss combines the advantages of both MSE loss and Mean Absolute Error (MAE). It behaves like MSE when the error is small and like MAE when the error is large, making it less sensitive to outliers. This loss function is defined in Eq. (3):

$$Huber = \begin{cases} 0.5(y-\tilde{y})^2, & |y-\tilde{y}| < \delta \\ \delta(|y-\tilde{y}| - 0.5\delta), & |y-\tilde{y}| \geq \delta \end{cases} \tag{3}$$

where δ controls the transition between the two behaviors. In the experiments, δ is set to 0.1.

VI. Score Calibration

The ConvNeXt models shown in Fig. 1 exhibit a tendency to predict MOS values biased toward the center of the scale. To mitigate this effect, a score calibration stage is applied individually to each model in the ensemble.

This stage expands the predicted MOS range using a parametric sigmoid-based transformation that compensates for the central bias observed during training and cross-validation. The parameters of the sigmoid function are optimized separately for each model, allowing it to adapt its output range more precisely and improving the ensemble's overall alignment with the true MOS distribution.

Fig. 5 shows the sigmoid functions corresponding to four trained ConvNeXt models. It is worth mentioning that these functions were optimized using the Out-Of-Fold (OOF) cross-validation samples, as described in the next section.

VII. Experiments and Results

The deep learning models were trained on a dataset of 288 filtered images [3]. Due to the limited size of the data provided for this challenge, the training and cross-validation process was carefully designed to achieve better generalization.

A. Experimental Setup

An 8-fold cross-validation scheme was implemented, partitioning the data into eight subsets. In each iteration, seven folds were used for training, while the remaining OOF served as the validation set. This procedure was repeated eight times, ensuring every fold acted as the OOF validation set once.

When training data is limited, employing a higher number of folds helps maximize the amount of data used for training in each fold iteration. However, this comes at the expense of higher computational cost, longer training durations, and a more complex overall pipeline.

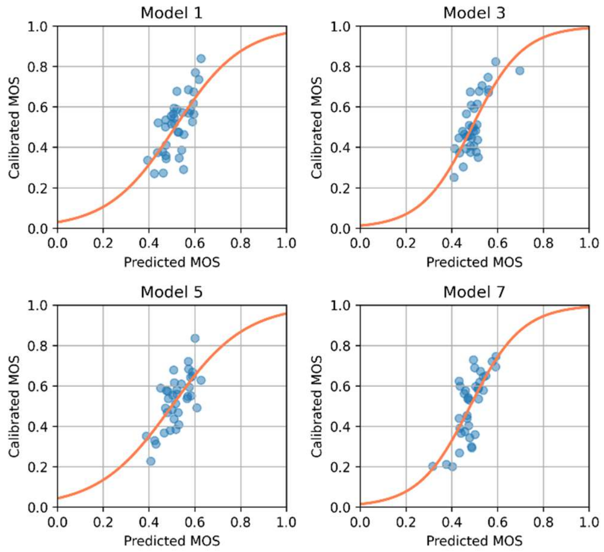

Fig. 5. Examples of sigmoid calibration functions for ConvNeXt models

A further difficulty with small datasets (for example, 288 images) is the risk of overfitting. This risk can be reduced by applying regularization techniques such as data augmentation and incorporating a dropout layer in the MLP head, as well as using a Cosine Annealing Learning Rate (LR) scheduler with Warm Restarts every 10 epochs (CAWR10). The latter proved particularly effective in reducing overfitting in this challenge.

Ten different training configurations were created by combining the described preprocessing techniques, regression heads, and loss functions. For each fold, one model was trained using each configuration, resulting in eight models per configuration. Each model was trained for 70 epochs using the AdamW optimizer, a batch size of 4, and the CAWR10 scheduler to anneal the learning rate from $1e^{-5}$ to $1e^{-7}$.

Training was conducted on a workstation powered by an Intel Core i9-9900K CPU and an NVIDIA GeForce RTX 4070 GPU. For each configuration n and fold k, two checkpoints were selected: one that minimized the validation loss and one that maximized the coefficient of determination (R^2). These checkpoints were then evaluated based on the average of the Pearson Linear Correlation Coefficient (PLCC) and the Spearman Rank-Order Correlation Coefficient (SROCC) between the ground-truth and OOF predictions. The best model for fold k was defined as the checkpoint that achieved the highest PLCC-SROCC average across all ten training configurations.

For comparison purposes, the reference implementation [5] of the baseline described in [3] was also trained using the same 8-fold cross-validation scheme, following its default configuration: input image size of 480×270 pixels, 80 training epochs, Adam optimizer, batch size of 4, MSE loss, an initial LR of $5e^{-6}$ with a decay factor of 0.9, and early stopping.

B. Experimental Results

Table I presents the training configurations that produced the best-performing model for each of the eight cross-validation folds. Each row corresponds to the configuration that achieved the highest PLCC-SROCC average in its respective fold, based on the evaluated checkpoints.

TABLE I. Best Training Setup per Cross-Validation Fold

Fold No.	Augmen-tation	Trans-formation	Reg. Head	Scheduler	Loss Fn.
1	Yes	No	MLP	CAWR10	Huber
2	No	Yes	Linear	CAWR10	Huber
3	No	No	Linear	CAWR10	WMSE
4	Yes	No	MLP	CAWR10	Huber
5	No	Yes	Linear	CAWR10	MSE
6	No	Yes	Linear	CAWR10	Huber
7	No	No	Linear	CAWR10	WMSE
8	Yes	No	Linear	CAWR10	Huber

Out of the ten training configurations tested, four unique configurations were ultimately selected, with some appearing in multiple folds. This variability highlights the complexity of the problem given the limited dataset and underscores the importance of evaluating multiple setups to improve predictive accuracy.

Table II presents a comparison of the baseline and proposed methods on the OOF sets, evaluated in terms of PLCC, SROCC, and Root Mean Squared Error (RMSE). The results are reported per fold, as an average across folds, and globally after concatenating the OOF data points.

TABLE II. Comparison of Baseline and Proposed Methods on OOF Sets Using PLCC, SROCC, and RMSE

	Baseline			Proposal		
Fold No.	**PLCC**	**SROCC**	**RMSE**	**PLCC**	**SROCC**	**RMSE**
1	0.608	0.609	0.115	0.695	0.658	0.100
2	0.435	0.447	0.131	0.647	0.585	0.110
3	0.514	0.445	0.118	0.757	0.622	0.089
4	0.673	0.573	0.106	0.672	0.592	0.105
5	0.489	0.471	0.127	0.666	0.691	0.107
6	0.466	0.558	0.134	0.620	0.608	0.117
7	0.503	0.307	0.114	0.697	0.634	0.091
8	0.628	0.563	0.123	0.735	0.680	0.107
Ave.	0.539	0.497	0.121	0.686	0.634	0.103
Glob.	**0.543**	**0.516**	**0.121**	**0.690**	**0.654**	**0.104**

As shown, the proposed model outperforms the baseline, achieving a PLCC of 0.690 compared to 0.543, an SROCC of 0.654 versus 0.516, and an RMSE of 0.104 versus 0.121.

Fig. 6 illustrates the relationship between the predicted and actual MOS for both approaches on the OOF sets. As observed, the proposed model's scatter plot exhibits fewer outliers than the baseline's, demonstrating its enhanced PLCC and SROCC metrics.

When applied to the test dataset comprising 72 unlabeled samples, the predicted MOS distributions from the two models are shown in Fig. 7. The proposed IQA method produces a wider distribution than the baseline, suggesting that it better captures variability in the data. This may lead to more accurate predictions, particularly at the lower and upper ends of the score range, which were underrepresented during training. The achieved PLCC and SROCC values are 0.627 and 0.589, respectively.

VIII. Conclusions and Further Work

This paper introduced a no-reference IQA model for predicting MOS in images processed with aesthetic filters. Owing to its strong performance and reliability for this particular task, the ConvNeXt-Large network was selected as the core component of the ensemble architecture.

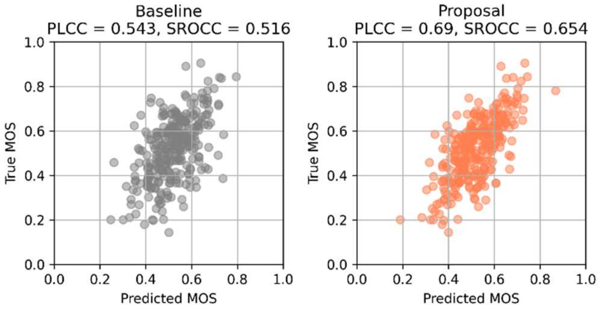

Fig. 6. Correlation between predicted and true MOS on the OOF sets for the baseline (left) and proposed (right) models

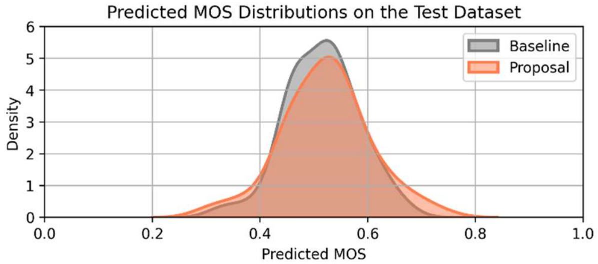

Fig. 7. Comparison of predicted MOS distributions on the test dataset

Special attention was given to the training process, where carefully selected configurations and tailored data pre- and post-processing strategies were applied to fully leverage the learning capabilities of the model. Furthermore, the proposed ensemble pipeline enhanced prediction robustness, compensating for the limited amount of training data.

Experimental results prove that the proposed pipeline outperforms the baseline, which is also based on ConvNeXt-Large. However, its deployment might be impractical in some scenarios due to the large size of the models (754 MB each).

Several strategies are proposed to reduce the model's complexity. One approach is to retain only the top-performing models in the ensemble. Another is to train compact models—e.g., ConvNeXt-Small (192 MB)—via knowledge distillation [8], a technique in which a smaller, lightweight "student" model learns to mimic a larger, pretrained "teacher" model.

References

[1] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a Practical Perceptual Video Quality Metric," Netflix Tech Blog, 2016.

[2] Z. Wang, E.P. Simoncelli, and A.C. Bovik, "Multi-scale Structural Similarity for Image Quality Assessment," Proc. 37th Asilomar Conf. Signals, Syst. Comput., Pacific Grove, USA, vol. 2, pp. 1398-1402, 2003.

[3] X. Wu *et al.*, "Image Manipulation Quality Assessment," IEEE Transactions on Circuits and Systems for Video Technology, vol. 35, no. 4, pp. 3450-3461, April 2025.

[4] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), New Orleans, USA, pp. 11976–11986, 2022.

[5] P.L. Rosin, H. Liu, J. Liu, Y. Liang, Y. Li, Y. Ma, X. Wu, and W. Zou, "Image Manipulation Quality Assessment Challenge," VCIP 2025, 2025. [Online]. Available: https://jiangliu5.github.io/imqac.github.io/

[6] PyTorch, "Model and Pre-trained Weights," PyTorch website. 2025. [Online]. Available: https://docs.pytorch.org/vision/main/models.html.

[7] A. Dudovskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," Proceedings of the International Conference on Learning Representations (ICLR), 2021.

[8] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," Proc. NIPS Deep Learning and Representation Learning Workshop, Montreal, Canada, 2014.