deimeq – A Deep Neural Network Based Hybrid No-reference Image Quality Model

Steve Göring, Alexander Raake
Audiovisual Technology Group; Technische Universität Ilmenau, Germany
Email: [steve.goering, alexander.raake]@tu-ilmenau.de

***Abstract*—Current no reference image quality assessment models are mostly based on hand-crafted features (signal, computer vision, ...) or deep neural networks. Using DNNs for image quality prediction leads to several problems, e.g. the input size is restricted; higher resolutions will increase processing time and memory consumption. Large inputs are handled by image patching and aggregation a quality score. In a pure patching approach connections between the sub-images are getting lost. Also, a huge dataset is required for training a DNN from scratch, though only small datasets with annotations are available. We provide a hybrid solution (deimeq) to predict image quality using DNN feature extraction combined with random forest models. Firstly, deimeq uses a pre-trained DNN for feature extraction in a hierarchical sub-image approach, this avoids a huge training dataset. Further, our proposed sub-image approach circumvents a pure patching, because of hierarchical connections between the sub-images. Secondly, deimeq can be extended using signal-based features from state-of-the art models. To evaluate our approach, we choose a strict cross-dataset evaluation with the Live-2 and TID2013 datasets with several pre-trained DNNs. Finally, we show that deimeq and variants of it perform better or similar than other methods.**

***Index Terms*—Image analysis, Machine learning, Image quality**

I. Introduction

Today digital images are everywhere. With smartphones and their built-in cameras as well as a variety of further entry-level up to professional digital cameras, million[1] pictures get uploaded to social media sites such as Instagram, Flickr or Facebook. Video and image quality assessment methods are key factors in building new compression algorithms to reduce file size, or to assess the quality-impact due to technical factors related with the employed camera system. Especially users want to get the best quality in every possible scenario, e.g. in rural areas with mobile connection or a fixed connection.

Also for robust and highly accurate pixel-based video quality models, it is important to have good performing underlying image quality models. For example, Netflix's VMAF [26, 18] is based on several image quality models. Considering, that in real world applications, a user or researcher will not have access to a reference video or image. This is why no-reference video and image quality assessment methods are getting more and more important. No-reference image quality models are mostly based on hand-crafted features (signal-based, computer-vision-based,...) [23, 14] or deep neural networks (DNNs) [13, 5, 1]. DNNs are already successfully used to address several image-related problems, for example classification [34], segmentation, face-detection, and more.

Especially for quality prediction, most existing models use a patching approach to avoid large input image sizes to DNNs (e.g. [13, 5]), because such large sizes result in large processing time or high memory consumption. The provided patching solution divides the input image into several smaller blocks and calculates, for each block, a quality score that later is aggregated to a final overall score. A pure patching-based approach leads to the problem that connected image portions and distortions are lost and therefore are not considered by the model. Furthermore, for training a new DNN from scratch, a huge human-annotated database is required.

A number of quality-annotated image datasets exists, for example Live-2 [30, 31] and TID2013 [28]. They only consider a relatively small number of images (e.g. TID2013: 25; Live-2: 29) with several distortion types, to finally provide 800-3000 distorted images. Another image dataset is KonIQ-10k [17], that does not include quality distortions like TID2013 or Live-2. Comparing to other image problems such as image classification (e.g. ImageNet competition: 150,000 images [29]), these databases are relatively small. However, a full re-training is not required in every use case, for example using transfer-learning.

A pre-trained DNN could be used as basis for re-training to a different problem space. In turn, it is hard to include in such a re-training process other, e.g. quality related, feature values without changing the complete DNN.

Our general idea – called *deimeq* – is to build an image quality model using a pre-trained DNN as automatic feature extractor. Instead of a pure patching approach, we are using hierarchically created sub-images, where the final features are aggregated. Later, the features are passed to a random forest model with feature selection. Before this step, the DNN features are optionally extended by state-of-the-art no-reference features. The name *deimeq* of our model stands for **de**ep **im**age no-reference hybrid model to **e**stimate **q**uality.

We will analyze the following identified research questions in our paper. Firstly, which pre-trained DNN can be used in combination with hierarchical sub-images for quality prediction? Secondly, which performance compared to other state-of-the-art models can be achieved using *deimeq*? Thirdly, will a combination of state-of-the-art no-reference features with our DNN-based features lead to a better overall prediction performance?

[1] for Flickr: average 1.63 million photos per day for 2017, see [7]

The paper is organized as follows: In Section II, a brief overview of the state-of-the-art image and video quality models is given. We discuss the main differentiating aspects of our model compared to the related work. Further, we describe our overall architecture in more detail in Section III. We conducted several experiments for evaluation of our proposed model, as described in Section IV, using a cross-dataset approach. Finally, we conclude with a discussion on the model and provide a short outlook and ideas for future work in Section V.

II. Related Work

Many full-reference, no- or reduced reference image or video quality models have been described in the current literature. We will focus on no-reference pixel-based image- or video-quality metrics that either use hand-crafted features, are related to deep neural networks or use other machine learning techniques.

A. Models using hand-crafted features

Netflix's VMAF metric achieved quite good results in predicting human ratings [26, 18]. VMAF is a compound video quality metric, it consists of several full-reference metrics and a per-frame motion estimator. Using these features, a Support Vector Machine (SVM) is trained to learn weights for calculating a combined quality score. VMAF is a full-reference video quality model. As several other more recent models, it uses a machine learning algorithm for final aggregation of the quality scores.

Besides video quality models, also several no-reference image quality models use a machine-learning-based feature integration. For example, brisque [23] consists of luminance-based scene statistics features that quantify possible losses of naturalness, and the SSEQ [20] model uses entropy-based spatial and spectral features (block- and DCT-based). For both models the underlying features are combined using a support vector regressor or similar regression algorithm to derive the final quality score.

Several other no-reference models exist that use hand-crafted features in combination with machine learning approaches [19, 24, 25]. For example, Liu et al. [19] introduces a feature set based on gradient orientations that is finally combined using a neural network. Currently, deep neural networks are used in several imaging applications such as classification or segmentation, where they show a good performance.

B. Models using DNNs

Hand-crafted features need to be modified or re-created if new distortions or new technologies appear. DNNs do not need such features and can take images as direct input. They were already successfully applied in a number of studies to still-image quality prediction [21]. For instance, Kang et al. describes a no-reference image quality metric based on a convolutional neural network using patches of 32x32 pixels for images of 512x768 resolution [13]. Similarly, Dash et al. also uses patches (64x64 pixels) to train a DNN. In their experiments, they achieved good regression results [5], for example reaching an accuracy of 98% using the CSIQ dataset [15]. However, in many state-of-the-art papers, only results for a cross-validation using the same databases are reported.

Another trend for DNN-based image quality models is the type of DNN used. For example, the VeNICE model [6] uses a pre-trained DNN (VGG16 [32]) with a patch size of 32x32 pixels. This shows that re-training from scratch is not required for quality prediction.

Furthermore, there are other similar approaches for no-reference image quality estimation using patches in combination with DNNs [16, 35]. Wiedemann et al. are focusing on a two steps approach, prediction importance of patches and later aggregating a final quality score using DNNs. In addition, approaches to image or video quality prediction without pure patching have been reported in the literature as well [2, 11]. For example, Göring et al. [9] are using a hierarchical sub-image creation approach combined with a pre-trained DNN to estimate VMAF quality scores as the first type of ground-truth data towards a no-reference video quality model trained and validated on video quality tests with users. To this aim, they apply the InceptionV3 DNN [34] for feature extraction, addressing encoding artifacts for the use-case of 4K video quality prediction.

C. Summary and identified key aspects

Considering the current state of the art image quality models, a few observations can be made. No-reference image-quality models are typically based on machine learning, such as SVMs, SVRs or random forest models, or rely on DNNs, mostly employ patching, and sometimes apply other approaches than patching. However, good performing no-reference models still use hand-crafted features in combination with regression algorithms to calculate the final score.

Current DNN-based models use small patches and are trained on low-resolution images, to avoid larger processing time. Patching does not preserve the global connections of distortions. Designing and re-training a DNN from scratch is a complex process, regarding processing time and input-parameter variability. Moreover, such a re-training also requires a large human-annotated database.

Furthermore, most approaches use a cross-validation approach in their papers. Instead, we will focus our approach on a cross-dataset evaluation, where completely separate datasets are used for validation. We also use complementary new distortion types in our validation step, to verify the general performance of our model. This circumvents the problem of distortion-specific features or models.

Our *deimeq* model will focus on the identified aspects and problems as described above. To tackle these, first, *deimeq* uses hierarchically sub-image creation and processing, second a pre-trained DNN as feature extractor is employed and last, we are able to extend the generated features with further no-reference image quality methods.

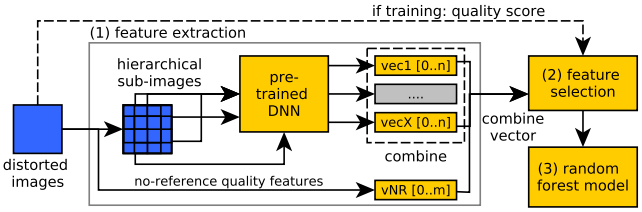

Fig. 1. General model structure of *deimeq*; a pre-trained DNN is used together with no-reference features to train a final model component with feature selection. For each input image, hierarchically created sub-images are used.

III. Proposed Model Architecture

In the following Section we describe in detail how *deimeq* works. The system uses a two-step training and validation approach. In the first, training step, the model is trained using a given feature set and training image database. The second step is the validation part, where the pre-trained model is applied to unknown images. Figure 1 shows the general model structure of *deimeq*, consisting of three main steps: (1) feature extraction with summarization to avoid huge dimensions and extension with state-of-the-art no-reference model features, (2) feature selection and (3) training of machine learning model.

A. DNN feature extraction and summarization

A pre-trained DNN is used as feature extractor. Each input image is divided into several sub-images in a hierarchical manner. All generated images are then rescaled to the input size of the DNN and processed. With this hierarchical approach, the smallest patch size to be chosen for a given model implementation depends on the input-image resolution of the underlying pre-trained DNN model. The smallest patch size is chosen so that the respective sub-image is just not downscaled (i.e. it is upscaled or preserved in size) when using it as input to the DNN. For example, let input images be of resolution $w_i \cdot h_i$ (with width w_i and height h_i of the input image) and the expected input resolution of the pre-trained DNN model be of $w_D \cdot h_D$ (with w_D the image width and h_D the image hight expected by the DNN). Then, to preserve optimal image resolution under the constraint of the DNN input, the hierarchical patching should contain at least l levels, with $l = \max\left(\log_2\left(w_i/w_D\right), \log_2\left(h_i/h_D\right)\right)$. Then, the smallest patches will just not be downscaled before input, fitting the requirement of maintained maximal resolution stated above. Based on these considerations, and as a result of the chosen image databases (TID2013 [28] and Live-2 [30], see Section IV) and selected pre-trained DNNs (see Section IV-B), we use the full images and sub-images with half of each dimension, i.e. $l = 1$. For higher input image resolutions, more levels are required. Besides preservation of input image resolution at the smallest patch sizes, this hierarchical approach ensures a connection between the distorted patches.

We use modified classification DNNs as basis, see Section IV for all DNNs, in the following we will use the Xception network [3] as an example. The modification is that we are not including the last – classification-oriented – layer (this is mostly a fully connected layer), and apply an average pooling strategy on the prev-last layer (it consists mostly of several identical parallel layers, e.g in case of Xception 5 layers with each 2048 values). Such a classification network would generate, e.g. in case of the Xception network, 2048 values for each sub-image. Using all theses values would lead to a huge dimensionality, which is why we apply a simple summarization of each of the generated prediction values of the DNN, assuming that the features would be similar in the sub-images, we therefore get only one vector f containing 2048 values.

The generated feature vector f is sparse, that means it includes many zeros. Because of that, we created a second vector $f_{!0}$, containing only the values that are not zero. As next, we calculate, for the generated feature vector f and the non-zero version $f_{!0}$, the following statistical values as vector s: mean, sum, standard deviation, skewness, kurtosis, harmonic mean (only for $f_{!0}$), geometric mean (only for $f_{!0}$), interquartile range and entropy.

These values together with f are a statistical description of the feature vector and are extended by one value that is the fraction of zeros in the feature vectors $1 - |f_{!0}|/|f|$ as an indicator how sparse the feature vector is. In case of the Xception [3] DNN we are calculating in sum 2065 values (2048 for f and the remaining values are based on the statistical values s) for each image.

The total number of our generated features is quite high in comparison with other state-of-the-art no-reference metrics, however due to the fact that we use pre-trained DNNs, we are not able to name and manually select the extracted features. For this reason, to reduce the overall calculation and dimensionality, we use automatic feature selection. Our feature selection step in the overall model pipeline will force to remove values that are not relevant.

B. Extension of features

Due to the fact that we are not re-training a pure DNN, we are able to extend our features with state-of-the-art no-reference values from other models if needed. We will analyze the extension with other features in our Section IV. We focus on the brisque [23] and niqe [22] features. Both features are luminance-based and perform combined quite well, also in comparison to other state-of-the-art models such as VeNICE [6].

Further, we will use re-trained variants of the additional NR-models of both feature sets as comparison baseline models. These re-trained models are based on the same feature-selection and random forest pipeline that we use for *deimeq*. With this re-training, we ensure that the baseline model performance is the best possible. In this step, also other no-reference quality/image features could be introduced.

C. Random forest model and feature selection

Our last step consist of a feature selection and training of a random forest model. The feature selection step uses a ExtraTreesRegressor using $0.001 \cdot mean$ as threshold for feature

importance selection. For all generated models we use 100 decision trees in our random forest model with mean squared error (MSE) as split criterion. All other parameters are default values provided by the used scikit-learn framework [27].

D. Further parameters or algorithms

During the selection process for the final modeling step, we also evaluated other machine learning algorithms (support vector regression, gradient boosting trees, ...) and model parameters. Here, the random forest and feature selection step and settings presented in this paper were found to perform as best and showed to be the most robust regarding all analyzed databases. However, our general idea is not restricted to these algorithmic choices or settings. Other combinations are suitable and will probably perform with similar results. *deimeq* is a meta concept consisting of the idea to use (i) a pre-trained DNN, (ii) with hierarchical sub-images and (iii) additional features to predict image quality using machine learning algorithms. It is also possible to apply such an approach to other image-related problems that are a focus of our current and future research.

IV. Evaluation

To evaluate our proposed method we use two databases, the Live-2 database [30] and the Tampere Image Database (TID2013) [28], in a cross-dataset evaluation approach. Using a cross-database evaluation will ensure that our provided model is not over-fitting to a specific database, and additionally it shows that the model is also able to perform on completely unknown data. Furthermore, for our *deimeq* model, we analyze different pre-trained DNNs in comparison to re-trained no-reference models – brisque and brisque+niqe. We also check if extending our model with these no-reference features will lead to a higher prediction accuracy. As metrics for evaluation of model performance we use several correlation values (Pearson, Kendall, Spearman) and the root mean squared error (RMSE).

A. Datasets

Before we evaluate our model, we will describe shortly the used image quality datasets. For evaluation, we used the Live-2-database [30] and TID2013 [28]. More image quality databases are available, e.g. CSIQ[15] or KonIQ-10k[17]. However, they are with lower resolutions, consisting less, only gray images or focus not on distortion oriented quality.

In our evaluation, we focus on Live-2 and TID2013, because they include similar distortion types and have approximately the same number of source images.

In Table I all key properties of both databases are summarized. The Live-2 dataset consists of 29 source images, in contrast to the 25 images of TID2013. TID2013 includes approximately 5 times more distortion types, therefore the total number of distorted images is approximately three times higher. Both datasets share some similar distortions. The image resolution for both datasets is relatively small for today's image sizes. However, for proving the effectiveness of our image quality modeling approach, the considered databases are practically useful. In the future, we will extend our evaluation to further public databases with other distortion types or higher resolution images once these may become available.

TABLE I
Image Quality Assessment Datasets

	Live-2	TID2013
# source images	29	25
# distortion types	5	24
# total distorted images	779	3000
image resolution (mostly)	768x512	512x384
quality score min/avg/max	0/51.5/100	3.4/62.1/100

For both datasets we transformed the published quality scores to the same scale. To this aim, we normalized the quality scores ([0,100]-DMOS in case of Live-2; [0-10]-MOS in case of TID2013) to a [0,100]-score using a linear mapping approach, where 0 is the lowest and 100 is the best quality score.

B. Performance of deimeq model variants

We analyze different pre-trained DNNs: Xception [3], VGG16 [32], VGG19 [32], ResNet50 [10], InceptionV3 [34], InceptionResNetV2 [33] and MobileNet [12]. All used DNNs are classification networks trained for the ImageNet competition [29].

Our implementation uses the keras framework [4] for DNNs and scikit-learn [27] for machine learning models. For the *deimeq* model variants and *brisque/niqe* models we checked and tuned the number of trees and feature selection (see Section III-C) of our pipeline. Changes of these parameters will just lead to minimal performance improvements.

In all experiments we trained on the Live-2 database and validated with the TID2013 images.

For a detailed analysis we trained several variants and calculated for each model performance metrics.

Considering Table II, only *deimeq* variants with Xception, InceptionV3 or ResNet50 DNNs are able to outperform the baseline *brisque/niqe* model variant. All other DNNs are not suitable in our setup, concluding that they are not reflecting quality-related features in their layers.

Our best performing model is *deimeq+*, using the Xception network in combination with *brisque* features. The performance of *deimeq* using *brisque* and *niqe* features is approximately the same as for *deimeq+*. We are able to get an approximately 10% higher Pearson-correlation with our *deimeq+* variant. A similar performance boost of the other correlations and the RMSE can be observed. In contrast, using only the DNN provided features without extension of no-reference features – *deimeq**, we are getting an approximately 6% higher correlation than the individual baseline models. There are also other models listed with similar performance. For example, *deimeq* with InceptionV3 shows similar performance regarding correlation and error rate.

Furthermore, the performance of the VGG16, VGG19, InceptionResNetV2 and MobileNet was worse than the base-

TABLE II
PERFORMANCE OF *deimeq* MODEL VARIANTS AND *brisque*, *P*=PEARSON, *K*=KENDALL AND *S*=SPEARMAN CORRELATIONS AND RMSE VALUES; **B**=BRISQUE/**N**IQE AS ADDITIONAL FEATURES; SORTED BY CORRELATIONS; TRAINED ON LIVE-2 AND VALIDATED WITH TID2013

model	used dnn	+feat.	*P*	*K*	*S*	RMSE
deimeq+	xception	**B**	0.53	0.32	0.47	17.33
deimeq	xception	**B+N**	0.53	0.32	0.46	17.04
deimeq	inceptionV3	**N**	0.53	0.28	0.40	15.99
deimeq	inceptionV3	**B**	0.52	0.33	0.47	17.69
deimeq	inceptionV3	**B+N**	0.52	0.31	0.45	17.35
deimeq	resnet50	**N**	0.52	0.30	0.43	16.77
deimeq*	xception		0.51	0.27	0.40	19.43
deimeq	inceptionV3		0.51	0.27	0.40	16.61
deimeq	resnet50	**B**	0.50	0.32	0.47	17.15
deimeq	xception	**N**	0.50	0.27	0.38	17.82
deimeq	resnet50	**B+N**	0.49	0.32	0.47	17.33
brisque			0.48	0.31	0.44	18.92
brisque		**N**	0.48	0.30	0.44	18.48
deimeq	vgg19	**B+N**	0.48	0.30	0.43	17.66
deimeq	vgg16	**B**	0.48	0.29	0.42	18.47
deimeq	vgg16	**B+N**	0.48	0.29	0.42	18.26
deimeq	incept-res	**B+N**	0.48	0.27	0.40	18.26
deimeq	vgg19	**B**	0.47	0.30	0.43	18.03
deimeq	mobilenet	**B**	0.47	0.30	0.43	17.54
deimeq	mobilenet	**B+N**	0.47	0.28	0.41	17.77
deimeq	incept-res	**B**	0.46	0.26	0.38	18.50
deimeq	resnet50		0.44	0.27	0.40	21.12
deimeq	mobilenet	**N**	0.41	0.20	0.30	18.18
deimeq	incept-res	**N**	0.41	0.19	0.28	20.23
deimeq	vgg19	**N**	0.38	0.17	0.25	19.60
deimeq	vgg16	**N**	0.36	0.17	0.24	21.02
deimeq	vgg19		0.36	0.14	0.21	26.17
deimeq	vgg16		0.29	0.13	0.19	26.40
deimeq	mobilenet		0.27	0.11	0.16	25.34
deimeq	incept-res		0.25	0.11	0.17	24.89

line models. Here, it needs to be considered that the used DNNs were originally designed for image classification tasks, a completely different image-analysis problem. Furthermore, some of the models are optimized for specific applications. For example, MobileNet was optimized for speed. To allow that it can be used in mobile applications, the layers are reduced and the overall model is smaller. Those properties can lead to a DNN that is usable for the specific problem area that it was trained for, however the usability as feature extractor for other applications decreases.

C. Further analysis

As an addition, we further analyzed the performance of all *deimeq* variants in comparison with *brisque* trained on TID2013 and evaluated on the Live-2 database, *deimeq* with Xception (Pearson=0.71, Spearman=0.72, Kendall=0.52) had similar correlation to *brisque* (Pearson=0.72, Spearman=0.73, kendall=0.53). In this setup the *brisque* model performs quite good, however it was developed specifically using the Live-2 database. Similar combinations of DNNs and no-reference features as in the Live-2-trained approach performed similar to the *brisque* baseline model. However, in this setup the TID2013 dataset contains all distortion types, and Live-2 only a subset, an evaluation regarding the missing distortion types would be required. We also analyzed the performance of all of our models in a pure 10-fold cross setup for each databases, Live-2 and TID2023. Our model and the baseline model perform well (correlations of more than 0.8). Since all databases include highly similar images, a 10-fold cross validation is considered less meaningful in this context.

Another validation considered the LIVE In the Wild Image Quality Challenge Database [8] (LIVEWILD). This database consists of crowd-sourcing quality annotated uniqe images without artifical distortion. A 10-fold cross validation is in this case not ciritcal, however our general model approach has as focus different distortion types and levels. In Table III the performance of the top 10 models are summarized. Best performing *deimeq* variant is again *deimeq+*, and vairants with Xception DNN. Xception and InceptionV3 are the most suitable DNNs for this database. Furthermore, the brisque/brisque+niqe baseline model performance is comparable with *deimeq*.

TABLE III
TOP 10 PERFORMANCE OF *deimeq* MODEL VARIANTS AND *brisque*, *P*=PEARSON, *K*=KENDALL AND *S*=SPEARMAN CORRELATIONS AND RMSE VALUES; **B**=BRISQUE/**N**IQE AS ADDITIONAL FEATURES; SORTED BY CORRELATIONS; CROSSVALIDATION ON LIVEWILD

model	used dnn	+feat.	*P*	*K*	*S*	RMSE
deimeq+	xception	**B**	0.62	0.42	0.6	14.98
deimeq	xception	**B+N**	0.62	0.41	0.59	15.02
deimeq	inceptionV3	**B+N**	0.6	0.4	0.58	15.29
deimeq*	xception		0.6	0.4	0.57	15.32
brisque		**N**	0.6	0.39	0.57	15.38
deimeq	xception	**N**	0.6	0.4	0.57	15.4
brisque			0.59	0.39	0.56	15.44
deimeq	mobilenet	**B+N**	0.59	0.4	0.57	15.43
deimeq	inceptionV3	**N**	0.59	0.39	0.57	15.49
deimeq	inceptionV3	**B**	0.59	0.39	0.57	15.53

In general, we found out that our approach can successfully be used with several image classification DNNs, 3 out of 7 DNNs were suitable regarding overall prediction performance. Further, such a model, without using additional no-reference features, is able to outperform state-of-the-art models in a cross-dataset evaluation approach. We use this "hard" evaluation approach with between-database training and validation to show how well such a model can perform and to ensure that the models are not over-trained for a specific image database with specific distortion types.

Lastly, using some state-of-the-art no-reference features will further improve, in mostly all cases, the overall performance of a successful *deimeq+DNN* approach.

V. CONCLUSION

First, we checked the current state-of-the-art no-reference image and video quality models and identified the following key issues and research questions. First, most well performing models use hand-crafted features in combination with machine learning to aggregate a final score. Second, other state-of-the-art models use deep neural networks with local patching and final score aggregation. Furthermore, training and tuning a DNN can be a complex task, due to the fact that huge datasets are required and that a pure patching will loose a global connection of distortions.

We introduce a hybrid model, called *deimeq*, that uses a pre-trained classification DNN as feature extractor in combination with hierarchical sub-images. Our generated features were combined, summarized and optionally extended by state-of-the-art no-reference values. For final score estimation *deimeq* uses a random forest model with a previously applied feature selection step.

In several analysis runs we analyzed which DNNs are suitable with hierarchical sub-images for quality estimation and combined the extracted DNN features with state-of-the-art no-reference features. We found out that only 3 out of 7 pre-trained DNNs can be used successfully as feature extractor for image quality problems; the best performing DNN was Xception. Our feature extension approach showed that, for example using Xception, such DNNs can be used in combination with brisque features to boost overall model performance. We showed that our *deimeq* approach without no-reference features is also able to out-perform current state-of-the-art and re-trained no-reference models in a hard cross-dataset evaluation approach.

Further research work should validate our provided approach for image quality assessment databases with higher resolutions and a more diverse image set including for example image liking aspects. For such datasets, our model can easily be extended by some state-of-the-art image aesthetic features. It is also possible to extend our *deimeq* idea to a full-reference system. However, no-reference models are more interesting, because a reference is not provided in most real-world image-quality assessment scenarios.

Acknowledgment

This research work was partially funded by Deutsche Telekom AG, Germany.

VI. References

[1] S. Bosse et al. “Neural network-based full-reference image quality assessment”. In: *PCS, 2016*. IEEE. 2016, pp. 1–5.

[2] A. Bouzerdoum et al. “Image quality assessment using a neural network approach”. In: *ISSPIT*. IEEE. 2004, pp. 330–333.

[3] F. Chollet. “Xception: Deep Learning with Depthwise Separable Convolutions”. In: *CoRR* (2016).

[4] F. Chollet et al. *Keras*. https://github.com/keras-team/keras.

[5] P. P. Dash et al. “Deep Quality: A Deep No-reference Quality Assessment System”. In: *arXiv* (2016).

[6] P. P. Dash et al. “VeNICE: A very deep neural network approach to no-reference image assessment”. In: *ICIT, 2017 IEEE Int. Conf. on*. IEEE. 2017, pp. 1091–1096.

[7] flickr. *Upload statistics*. URL: https://www.flickr.com/photos/franckmichel/6855169886/ (visited on 06/15/2018).

[8] D. Ghadiyaram and A. C. Bovik. “Massive online crowdsourced study of subjective and objective picture quality”. In: *IEEE Trans. Image Process* 25.1 (2016), pp. 372–387.

[9] S. Göring et al. “DeViQ – A deep no reference video quality model”. In: *Electronic Imaging* (Jan. 2018).

[10] K. He et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* (2015).

[11] W. Hou et al. “Blind image quality assessment via deep learning”. In: *IEEE trans. on neural networks and learning systems* 26.6 (2015), pp. 1275–1286.

[12] A. G. Howard et al. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”. In: *CoRR* (2017).

[13] L. Kang et al. “Convolutional neural networks for no-reference image quality assessment”. In: *CVPR*. 2014, pp. 1733–1740.

[14] V. Laparra et al. “Perceptual image quality assessment using a normalized Laplacian pyramid”. In: *EI* 2016.16 (2016), pp. 1–6.

[15] E. C. Larson and D. M. Chandler. “Most apparent distortion: full-reference image quality assessment and the role of strategy”. In: *J Electron Imaging* 19.1 (2010), pp. 011006–011006.

[16] J. Li et al. “No-reference image quality assessment using Prewitt magnitude based on convolutional neural networks”. In: *Signal, Image and Video Processing* 10.4 (2016), pp. 609–616.

[17] H. Lin et al. *KonIQ-10K: Towards an ecologically valid and large-scale IQA database*. 2018.

[18] J. Y. Lin et al. “A fusion-based video quality assessment (fvqa) index”. In: *APSIPA, 2014 Asia-Pacific*. Dec. 2014, pp. 1–5.

[19] L. Liu et al. “Blind image quality assessment by relative gradient statistics and adaboosting neural network”. In: *Signal Processing: Image Communication* 40 (2016), pp. 1–15.

[20] L. Liu et al. “No-reference image quality assessment based on spatial and spectral entropies”. In: *Signal Proc.: Image Communication* 29.8 (2014), pp. 856–863.

[21] V. V. Lukin et al. “Combining full-reference image visual quality metrics by neural network.” In: *HVEI*. 2015, 93940K.

[22] A. Mittal et al. “Making a ”completely blind” image quality analyzer”. In: *IEEE Signal Process. Lett.* 20.3 (2013), pp. 209–212.

[23] A. Mittal et al. “No-reference image quality assessment in the spatial domain”. In: *IEEE Trans. Image Process.* 21.12 (2012), pp. 4695–4708.

[24] A. K. Moorthy and A. C. Bovik. “A two-step framework for constructing blind image quality indices”. In: *IEEE Signal Process. Lett.* 17.5 (2010), pp. 513–516.

[25] A. K. Moorthy and A. C. Bovik. “Blind image quality assessment: From natural scene statistics to perceptual quality”. In: *IEEE Trans. Image Process.* 20.12 (2011), pp. 3350–3364.

[26] Netflix. *Netflix VMAF*. URL: https://github.com/Netflix/vmaf (visited on 01/24/2018).

[27] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *JMLR* 12 (2011), pp. 2825–2830.

[28] N. Ponomarenko et al. “Image database TID2013: Peculiarities, results and perspectives”. In: *Signal Proc.: Image Communication* 30 (2015), pp. 57–77.

[29] O. Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *IJCV* 115.3 (2015), pp. 211–252.

[30] H. R. Sheikh et al. “A statistical evaluation of recent full reference image quality assessment algorithms”. In: *IEEE Trans. Image Process.* 15.11 (2006), pp. 3440–3451.

[31] H. Sheikh et al. *LIVE image quality assessment database release 2 [OL]*. 2006.

[32] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *CoRR* (2014).

[33] C. Szegedy et al. “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”. In: *CoRR* abs/1602.07261 (2016).

[34] C. Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”. In: *CoRR* (2015).

[35] O. Wiedemann et al. “Disregarding the Big Picture: Towards Local Image Quality Assessment”. In: *QoMEX*. Sardinia, Italy, May 2018.