Convolutional Neural Networks for Omnidirectional Image Quality Assessment: Pre-Trained or Re-Trained?

Abderrezzaq Sendjasni, Mohamed-Chaker Larabi, Faouzi Alaya Cheikh

CONVOLUTIONAL NEURAL NETWORKS FOR OMNIDIRECTIONAL IMAGE QUALITY ASSESSMENT: PRE-TRAINED OR RETRAINED?

Abderrezzaq Sendjasni [1,2], *Mohamed-Chaker Larabi* [1] *and Faouzi Alaya Cheikh* [2]

[1] CNRS, Univ. Poitiers, XLIM, UMR 7252, France
[2] Faculty of Information Technology and Electronics, NTNU, Norway

ABSTRACT

The use of convolutional neural networks (CNN) for visual quality assessment (VQA) has become many researcher's focus. Various pre-trained models have been fine-tuned and used for this task. In this paper, we conducted a benchmark study of 7 state of the art pre-trained models for VQA of omnidirectional images. To this end, we first trained these models using an omnidirectional database and compared their performances with the pre-trained versions. Second, we compared the use of viewports and equirectangular images as the inputs to the models. Then, for the viewports-based models, we explored the impact of their numbers on the models' performances. Experimental results demonstrated the performance gain of the trained CNNs.

Index Terms— Omnidirectional images, CNN, blind visual quality assessment

1. INTRODUCTION

In recent years, virtual reality (VR) applications has known impressive growth. Omnidirectional images (also known as 360-degree images) represent an important part of these applications' content, in which the users are provided with a visual experience of real-world scenes. The users get an immersive experience when using Head-Mounted Displays (HMD), where they are allowed to freely focus on the desired content. As the viewing context for this type of content is different compared to conventional 2D. The users may get a completely different experience in terms of perception and immersivity. And, with the large introduction of this type of content in our daily life, this requires to validate the quality of experience brought by the various applications, hence, visual quality assessment (VQA).

VQA is a well known challenge in image processing and computer vision. It refers to the process of measuring the weighted combination of all visual attributes that reflect the perceptual quality of a given image/video. This is performed in a way that is consistent with human subjective opinions. At present, VQA has been extensively studied for 2D content. However, for omnidirectional content, it is still in its childhood and not fully studied. As omnidirectional content is generally processed, encoded, and transmitted using a 2D plane representation, one can apply existing 2D quality metrics directly on it. Still, these metrics do not account for the non-uniform sampling density at pixel locations from sphere to plane projection [1]. And, their performances are lacking in terms of correlation with subjective quality scores as it was shown in [2]. Thus, quality metric for omnidirectional content is of a paramount importance.

Recently, the use of machine learning techniques in image processing has gained a lot of attention. Deep Learning is a sub-field of machine learning inspired by the structure and function of the brain called artificial neural networks. Convolutional neural network (CNN) is a class of deep neural networks applied to analyze visual imagery [3]. CNN models have achieved great performances in various image processing tasks compared to conventional methods including object detection, classification, and segmentation. The main power of a CNN lies in its architecture, which is capable to extract distinctive features at various levels of abstraction [4–7]. From low-level features like edges and colors to high-level features like faces and objects. Foreseeing its remarkable performance, CNN-based approaches for visual quality assessment have been proposed [8, 9] for 2D images and [10–12] for omnidirectional images. In these works, well known CNN architecture are used such as VGG [6] and ResNet [5] architecture. In particular, the authors in [8] proposed to extract patches from images based on scan-path fixation points as input to a CNN model. They adopted [6] model to fine-tune after a comparison of 4 models. In [9], the authors proposed a synthetic CNN (S-CNN) for synthetic distortions. They combined it with VGG-16 [6] using a bilinear pooling as their output shape are different. Despite the impressive results achieved by [8, 9], they are designed for 2D images. As for omnidirectional ones, in [10], the authors adopted the ResNet architecture, where they proposed a viewport based approach with a multi-channel CNN. First, they projected the equirectangular (ERP) image into six viewports. These viewports goes as inputs to six parallel ResNet-34 [5] that share the same weights. Finally, the outputs of the six channels go as input to an image quality regressor, which concatenates the extracted features and and drives a quality score. In [11], a viewport based approach is also proposed. Here, the authors pro-

posed to benefit from the spatial mutual dependencies among the extracted viewports. For that, they used a graph CNN. The authors also used the DB-CNN proposed in [9] to compute the global quality where the input is a down-sampled ERP image. Its output is combined with the output of the graph CNN to predict the quality score. In [12], the positions of selected patches are considered along with their content. A total of 32 patches are derived from each ERP image. The position features are fused with visual features to predict the quality score of the selected patches. ResNet-50 was used to extract visual features. The overall quality score of all 32 patches is then fused and passed to a perception quality guider. In the aforementioned works, each used a well known pre-trained model for different tasks, some as a part of their network, other as the base model. For example, as a viewport descriptor in [11] and as a feature extraction in [10, 12]. One reason is, these models have been trained in a very large database.

A major challenge in the case of omnidirectional images is the existence of a reliable and representative database [2] that would allow deep learning models to show their full potential. This makes the use of pre-trained models seems a better alternative to compensate for such lack. We called it transfer-learning (TL). But, is it wise to use pre-trained models for VQA? And which one is suited for this task?

In this paper, we attend to answer the above questions by conducting a benchmark using different CNN models. First, we generate viewports surrounding the equatorial line covering around 110° of the vertical field of view (FOV). This represents 60% of the content. Then, we evaluated the impact of different multiple inputs on the performances of pre-trained models. Here, each input goes to a pre-trained model resulting in multiple models in parallel. We varied the number of inputs from 4 to 24 including 8 and 16. Besides, we evaluated the performance of these models on ERP images. For this, we down-sampled the ERPs by a factor of 4 to the ratio and used them as inputs to the different models. Finally, we compared the models' behavior when used with their original weights and retrained from scratch. Withal, the provided discussions based on the evaluation results are intended to provide insights on the use of pre-trained models for omnidirectional image quality assessment.

2. THE PROPOSED METHOD

To cover most used pre-trained models applied in IQA, we selected four different CNN architectures, including ResNet, VGG, Inception, and DenseNet. The selected models were all trained on the well known ImageNet [13] database. A model trained on ImageNet has essentially learned to identify both low-level and high-level features in images as it contains a thousand samples. Besides, for specific applications such as VQA, CNN models have to be trained on data drawn from those applications. Unfortunately, This is time-consuming and in some cases not suitable due to a lack of sufficient training data as in the case of omnidirectional content. One solution is that a model trained on a large scale database can use its weights for other image processing tasks. This reduces the load of training from scratch and, smaller domain-specific training data may be sufficient. Therefore, in this study, we fine-tuned each of the selected models. The fine-tuning consists of adding a regression block on top of all models. This block includes a Global Average Pooling, fully-connected layer, dropout [14], and a regression layer. In the following sub-section, a brief description of each model's architecture is provided.

2.1. Pre-trained CNN models

ResNet : residual networks are artificial neural networks that were introduced in 2015 [5]. The ResNet utilize skip connections to jump over some layers. This helps training deeper network layers without falling into the problem of vanishing gradients. There are several variants of ResNet, including ResNet-18, ResNet-34, and ResNet-50. The numbers denote convolutional layers. All these variants are explored in this study.

VGG : VGG is a convolutional neural network architecture proposed in [6]. This network is characterized by its simplicity and only use 3×3 convolutional layers stacked on top of each other in increasing depth. All convolutional layers are divided into 5 groups and each group is followed by a max-pooling layer. There are different versions of this network, the Vgg-16 and Vgg-19 are considered in this study.

Inception : the inception network architecture is introduced in [7] by Google. This network is composed of inception modules. These modules are used in CNN to allow for more efficient computation and deeper Networks through a dimensionality reduction with stacked 1×1, 3x3, 5x5, and pooling convolutions. Several variations of this network also exist. Here we used the Inception-V3 model.

DenseNet : DenseNet is a neural network composed of Dense blocks [4]. In each block, the layers are densely connected. These connections mean that the network has $L(L+1)/2$ direct connections. L is the number of layers in the architecture. Each layer receives input from all previous layers' output feature maps.

To provide a comprehensive analysis of the use of pre-trained CNN, we consider two parts in this study. A viewport-based approach as in [10–12] and an ERP-based approach inspired by [9, 12]. Fig. 1 depicts the proposed benchmark's structure. In the following, details of each part are provided.

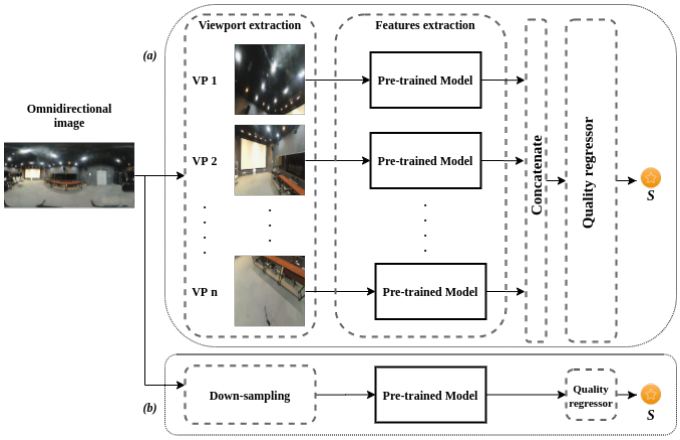

Fig. 1. Structure of the proposed benchmark.

2.2. Benchmark architecture

2.2.1. Viewport-based model

Inspired by the way the human gaze is biased towards the equatorial line when viewing omnidirectional images [15]. And, the fact that more than 30% of this content is not viewed. We generated viewports surrounding the equatorial line that represent 60% of the input content. To obtain the selected viewport $Vp_{i,i=1...k}$ with $k = 24$, we projected it spherical content to a 2D plane. Then, each Vp goes as input into a pre-trained model. The model architecture depends on the viewports' numbers. It consists of K models in parallel. Thus, the models' complexity is increased accordingly. In the end, the quality score is obtained by fusing the concatenated output feature maps. Fig 1(a) depicts the framework adopted. For the end to end training, we compute the error between predicted and target scores. For, this the mean squared error is adopted as loss function and it's defined as:

$$Loss = (q_{predicted} - q_{target})^2 \quad (1)$$

Where $q_{predicted}$ denotes the predicted score by the model and q_{target} in the mean opinion score (MOS). In this part, the training was performed on 80% of the generated viewports. Then, the trained model is used as a base model, see Fig.1(a). Here, we further trained the quality regressor module on the same set and tested the whole network on the other 20%. The idea adopted here is that each viewport inherits the MOS of its image which allows us to increase the database without altering the content. In total, the training set obtained is 10128 image with a $(256 * 256 * 3)$ shape.

2.2.2. ERP-based model

To evaluate CNN models on high-resolution ERP images. We used ERP images as input. Here, we down-sampled all images by a factor of 4, resulting in $1024 * 512$ of resolution. This implies that the models' input is also changed to match the shape of the input images. Fig 1(b) depict the structure of the proposed method. The same error function denoted by Equ.1 is used for the end to end training.

3. EXPERIMENTAL RESULTS

Database:

This study is carried out on the CVIQD2018 [16] database. It contains 16 original omnidirectional images, compressed using JPEG, H.264/AVC, and H.265/HEVC codecs. In total, it counts 528 distorted images which makes it the largest available database in this field. We split randomly the database into 80% (422 images) for training and 20% (106 images) for testing. For a fair comparison, all models were trained/tested using the same splitting scheme. The use of a second database is very important for model validation and generalization. Unfortunately, we couldn't acquire one.

Experimental setting:

The proposed benchmark is implemented using TensorFlow [17] and will be publicly available[1]. As we trained the models, the hyper-parameters we tuned were the optimizer algorithm, learning rate decay, batch size, and the number of layers to append so as to perform TL. We tried RMSProp, Adam and stochastic gradient descent (SGD) optimizer and settled on using RMSProp. For the learning rate decay, we adopted the ExponentialDecay starting from a $learningrate = 1e{-}4$. Then, we trained all models using mini-batch sizes of 3 for 25 epochs. We standardized the dimension of all models' input shape to $(256, 256, 3)$.

3.1. Performance evaluation

To assess the performance of selected models, we used four common metrics including Pearson Linear Correlation Coefficient (PLCC), Spearman Rank Order Correlation Coefficient (SRCC). A model is considered to have a better performance when PLCC and SRCC are close to 1. The performances given are computed on normalized data between $(0, 1)$. The predicted scores are fitted using five parameters non-linear logistic function as given below :

$$f(x) = \beta_1 \left(\frac{1}{2} - \frac{1}{1 + \exp^{\beta_2 (x - \beta_3)}} \right) + \beta_4 x + \beta_5 \quad (2)$$

where x denotes the objective score and, $f(x)$ represents the corresponding mapped score. $\beta_i (i = 1, 2, 3, 4, 5)$ correspond to the logistic function parameters to be fitted.

[1] https://github.com/sendjasni/360-IQA-CNN-BENCH

Table 1. Performance evaluation of pre-trained models using original weights. Best performing models are highlighted in bold for rows and underlined for columns

Models			ResNet-34	ResNet-18	ResNet-50	DenseNet-121	Vgg-16	Vgg-19	Inception-V3
Input type	Vp = 4	PLCC	0.899	0.932	0.945	**0.961**	0.907	0.935	0.942
		SRCC	0.885	0.900	0.909	**0.944**	0.875	0.909	0.913
	Vp = 8	PLCC	0.917	0.921	**0.953**	0.951	0.932	0.932	0.943
		SRCC	0.874	0.878	**0.930**	0.912	0.894	0.882	0.936
	Vp = 16	PLCC	0.938	0.920	**0.961**	0.955	0.947	0.934	0.937
		SRCC	0.921	0.880	**0.941**	0.920	0.916	0.888	0.889
	Vp = 24	PLCC	0.925	0.903	0.940	**0.960**	0.949	0.940	0.940
		SRCC	0.921	0.879	0.907	**0.930**	0.906	0.883	0.915
	ERP	PLCC	0.906	0.887	0.950	**0.954**	0.942	0.904	0.924
		SRCC	0.882	0.863	0.923	**0.932**	0.903	0.858	0.891

Table 2. Performance evaluation of retrained models. Best performing models are highlighted in bold for rows and underlined for columns

Models			ResNet-34	ResNet-18	ResNet-50	DenseNet-121	Vgg-16	Vgg-19	Inception-V3
Input type	Vp = 4	PLCC	0.974	0.977	0.979	**0.984**	0.965	0.981	0.981
		SRCC	0.954	0.957	0.966	**0.972**	0.951	0.965	0.970
	Vp = 8	PLCC	0.979	0.981	0.975	**0.982**	0.972	0.976	0.978
		SRCC	0.967	0.961	0.944	**0.976**	0.960	0.965	0.964
	Vp = 16	PLCC	0.977	0.980	**0.984**	0.983	0.976	0.982	0.979
		SRCC	0.961	0.968	**0.974**	0.976	0.962	0.973	0.972
	Vp = 24	PLCC	0.976	0.975	0.978	0.981	0.979	**0.983**	0.981
		SRCC	0.961	**0.972**	0.955	0.972	0.964	**0.972**	**0.972**
	ERP	PLCC	0.962	0.933	0.942	0.962	**0.947**	0.930	0.944
		SRCC	**0.918**	0.899	0.905	0.930	0.920	0.887	0.905

3.2. Results and discussion

With the aims to drive conclusions regarding which models best fit the quality assessment context, we compared 7 state of the art CNN models including ResNet-34, ResNet-18, ResNet-50, DesneNet-121, Vgg-16, Vgg-19 and Inception-v3. The performance's results are provided in Table 1 and Table 2. These performances are obtained on the testing set.

Overall, the performances are satisfactory. In particular, DenseNet-121 and ResNet-50 outperformed the other models in both viewport-based and ERP-based approach. Regardless of DenseNet-121 performance, the DenseNets models are quite neglected in VQA tasks, as most recent works adopted either ResNets or Vggs [8, 9, 12, 14]. Comparing the two approach explained in Sec.2.2, we observe that the performance of retrained models stood out compared to their pre-trained versions. From Table 2, all models achieved a correlation of 0.93 or higher when trained on ERPs and 0.96 or higher when trained on viewports covering the 110° vertical FOV. This may due to the fact that retrained models have learned the notion of quality after been trained to predict it. When using ERP images for training, the improvement is significant compared to pre-trained models with ERPs as inputs. Same behavior can be observed when retrained on viewports. Also, the strategy used for training could be the key, as we trained the models on more than 10000 samples. Between ERPs and viewport based inputs for pre-trained models, we can observe from Table 1 that, the former performed better as a trade-off between performances and complexity. Here, no omnidirectional peculiarities have been incorporated. Despite that, their performances are comparable with the viewport-based method which uses the important content on omnidirectional images. In Table 2, we observe a completely different behavior. Retrained models on extracted viewports outperformed by far their retrained versions on ERPs. This is because users don't watch the whole images. Their ratings are often based on a portion of this later. In terms of viewport inputs number, one can observe different behavior for the different models. For example, retrained models didn't significantly improved with increased inputs. Here, the complexity brought by the increasing of inputs could not be worth it. ResNet-50 and ResNet-34 performances improved with more inputs and slightly dropped with Vp=24. As for ResNet-18, the opposite is observed. In general, one can observe a performance saturation at Vp=16. Training the selected CNN models on omnidirectional images showed impressive results. The validation of these models' performances cannot be done without a second database. To provide representative results, we made

sure to test all models on unseen images represented by the 20%. As the diversity of content is necessary for models to generalize well, the results provided in Table1 and 2 are considered content dependent. As we trained all models for 25 epochs, some may have needed more, some less. Here, techniques such $EarlyStopping$ and $ModelCheckPoint$ may benefit to the models and avoid overfitting. This help to stop the training once no improvement is observed and before the model overfits. For this study, the choice of 25 epochs was made based on [8, 12, 14] for the sake of a fair comparison.

4. CONCLUSION

In this paper, we investigated the use of well known CNN models for VQA of omnidirectional content. Recent works have adopted different state of the art models originally designed for classification tasks. The reason behind this choice is that these models have been trained on significantly larger databases, here TL techniques may benefit VQA. As different models have been used, we studied 7 from a different aspect. We compared retrained models on omnidirectional content and compared their performances with their pre-trained versions. The performances achieved are statistically significant. We also considered the use of viewports as inputs and ERP images. The obtained results show that using ERPs best fits as a trade-off between complexity and performance gains for pre-trained models. For retrained ones, viewport-based training performed the best. We believe this study could bring insight into the use of pre-trained CNN models for VQA. A second database is in urgent need to validate the generalization of the proposed method based on CNN.

5. REFERENCES

[1] M. Yu, H. Lakshman, and B. Girod, "A framework to evaluate omnidirectional video coding schemes," in *IEEE International Symposium on Mixed and Augmented Reality*, Fukuoka, Japan, 2015, pp. 31–36.

[2] A. Sendjasni, MC. Larabi, and FA. Cheikh, "On the improvement of 2D quality assessment metrics for omnidirectional images," in *Electronic Imaging*, Burlingame, California USA, 2020, pp. 287–1.

[3] M.V. Valueva, N.N. Nagornov, P.A. Lyakhov, G.V. Valuev, and N.I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Mathematics and Computers in Simulation*, vol. 177, pp. 232 – 243, 2020.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 4700–4708.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 2016, pp. 770–778.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2818–2826.

[8] A. Chetouani and L. Li, "On the use of a scanpath predictor and convolutional neural network for blind image quality assessment," *Signal Processing: Image Communication*, vol. 89, pp. 115963, 08 2020.

[9] W. Zhang, K. Ma, J. Yan, D. Deng, and Z. Wang, "Blind image quality assessment using a deep bilinear convolutional neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 36–47, 2020.

[10] W. Sun, W. Luo, X. Min, G. Zhai, X. Yang, K. Gu, and S. Ma, "MC360IQA: The multi-channel CNN for blind 360-degree image quality assessment," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Sapporo, Japan, Japan, 2019, pp. 1–5.

[11] J. Xu, W. Zhou, and Z. Chen, "Blind omnidirectional image quality assessment with viewport oriented graph convolutional networks," *IEEE Transactions on Circuits and Systems for Video Technology (Early Access)*, pp. 1–1, 2020.

[12] H. G. Kim, H. Lim, and Y. M. Ro, "Deep virtual reality image quality assessment with human perception guider for omnidirectional image," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 4, pp. 917–928, 2020.

[13] O. Russakovsky, J. Deng, H. Su, J Krause, and S. Satheesh et al., "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[14] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[15] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein, "Saliency in VR: How do people explore virtual environments?,"

IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 4, pp. 1633–1642, 2018.

[16] W. Sun, K. Gu, S. Ma, W. Zhu, N. Liu, and G. Zhai, "A large-scale compressed 360-degree spherical image database: From subjective quality evaluation to objective model comparison," in *IEEE 20th international workshop on multimedia signal processing (MMSP)*, Vancouver, BC, Canada, 2018, pp. 1–6.

[17] A. Martín, P. Barham, J. Chen, Z. Chen, and A. Davis et al., "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation*, Savannah, GA, USA, 2016, pp. 265–283.