ROBUSTNESS AND PREDICTION ACCURACY OF MACHINE LEARNING FOR OBJECTIVE VISUAL QUALITY ASSESSMENT

Andrew Hines[⋆], *Paul Kendrick*[†], *Adriaan Barri*[‡], *Manish Narwaria*[§], *Judith A. Redi* [$]

[⋆]Trinity College Dublin, Ireland [†]University of Salford, UK [§]Université de Nantes, France
[‡]Vrije Universiteit Brussel and iMinds, Belgium [$]Delft University of Technology, the Netherlands

ABSTRACT

Machine Learning (ML) is a powerful tool to support the development of objective visual quality assessment metrics, serving as a substitute model for the perceptual mechanisms acting in visual quality appreciation. Nevertheless, the reliability of ML-based techniques within objective quality assessment metrics is often questioned. In this study, the robustness of ML in supporting objective quality assessment is investigated, specifically when the feature set adopted for prediction is suboptimal. A Principal Component Regression based algorithm and a Feed Forward Neural Network are compared when pooling the Structural Similarity Index (SSIM) features perturbed with noise. The neural network adapts better with noise and intrinsically favours features according to their salient content.

Index Terms— image quality assessment, SSIM, neural networks, machine learning

1. INTRODUCTION

Objective Visual Quality Assessment (OVQA) is an important module in the maintenance of an acceptable Quality of Experience level in multimedia delivery systems [1]. For instance, a video coding system requires the knowledge of video quality for appropriate bit allocation; similarly, post-processing chains in displays need to estimate the quality of the incoming video to calibrate and apply image restoration algorithms.

As natural signals, neighbouring samples of images and videos are correlated, and can be well approximated by a first order Markov process. Human visual perception mechanisms are well equipped to only retain the useful signal information, discarding the redundant information [2,3] for perceiving signal quality. However, incorporating such mechanisms into a mathematical automated quality prediction algorithm (hereafter referred to as metric) is challenging. This is primarily due to a limited understanding of the complex human perceptual mechanisms, and to the computational complexity that their existing models typically entail. As a result, an efficient metric that accurately mimics visual quality perception is yet to be found, despite existing efforts [4, 5].

Recently, Machine Learning (ML) has been proposed as a suitable tool to support OVQA [6, 7]. ML has been exploited as a promising data-driven image and video feature pooling strategy towards perceptual quality assessment, given that the exact pooling mechanisms of the human visual system are believed to be complex [8]. Using ML, accurate OVQA models of complex non-linear mechanisms have been developed in a computationally tractable way and based on a limited set of training examples [6], yet achieving high agreement with subjective ground truth [6, 8, 9]. However, prediction accuracy must be tempered against the risk of overspecialisation, caused by the high number of metric parameters set using the training examples [10]. As a result, the robustness of ML-based OVQA metrics is often questioned.

This paper does not assess either the accuracy or the robustness of specific ML-based OVQA metrics. It investigates the valuable intrinsic feature selection ability of ML. Features extracted from images or videos often carry either redundant or non-relevant information, making the feature space noisy, and consequently suboptimal, for accurate quality prediction. This study shows how ML techniques can achieve high prediction accuracy for quality estimation by filtering irrelevant information from suboptimal spaces.

Due to the availability of labelled training data with subjective ground truth quality assessments and clear benchmarks, this study focuses on objective Image Quality Assessment. In particular, it uses SSIM (Structural Similarity Index) [11] as a test tool for evaluating ML methods. SSIM is a widely used metric based on the pooling of three component features. These are particularly suitable for being used as proxies for input features which could be corrupted in a controlled manner. It must be stressed that other studies have demonstrated that ML can improve on the performance of SSIM, e.g. [12], whereas this is not the goal of the present paper. Here, SSIM and benchmarks are used to investigate the suitability of the use of ML for OVQA in the presence of suboptimal conditions. A controlled study on the ML performance with varied input feature adjustments is conducted. The performance of a *conventional* metric using a linear combination of input features is compared to a metric that uses a neural network [10] to combine the same set of features into an OVQA score. To simulate different levels of optimality with which the feature space captures the human visual system, noise is added directly to the SSIM component features.

The robustness of both metrics, based on increasingly noisy feature spaces, is tested in predicting the quality of the images in the LIVE database [13].

The remainder of this paper is organised as follows. Section 2 details the setup of the metrics involved in the comparison. Section 3 describes the addition of noise and the experimental setup. Section 4 reports and discusses the test results and is followed by concluding remarks in section 5.

2. MACHINE LEARNING FOR OBJECTIVE QUALITY ASSESSMENT

Given an image *i**, the goal of an image quality metric is to predict the quality score *q** perceived by the user observing *i**. Such prediction is typically accomplished by determining a set of F features meaningful for perceptual quality $\mathbf{f} = \{f_j(i*), j = 1, \ldots, F\}$ and then linking them to *q** through some function *m*. While in many cases the model *m* is established a priori, (e.g., [14]) ML techniques allow *m* to be determined in a data-driven way, i.e., based on a set of n_p observations $\{i_l, q_l\}, l = 1, \ldots, n_p$, such that:

$$m(\mathbf{f}(i_l)) = q_l + \epsilon_l, \tag{1}$$

where ϵ_l is the estimation error [6].

The design of quality metrics builds on a selection and characterisation of perceptual features (e.g. spatial, frequency or temporal), which are used to compute a predicted quality score (see eq. 1). SSIM, for example, combines three perceptually relevant features, related to luminance (mean intensity), contrast (variance) and structure (covariance) information, into a quality score [11]. Quality prediction is achieved by a multiplication of the features according to a predetermined model *m*.

There are drawbacks with such a pooling model. First, the functional form chosen (a parameterized multiplicative model) is a priori and may not be the optimal one. Second, there is no systematic way to determine the values of the 3 pooling parameters [11] (for the original SSIM implementation, all are equally weighted through parameters $\alpha = \beta = \gamma = 1$). These issues can be addressed using ML for feature combination by selecting *m* (and its parameters) via a training process that adaptively updates the configuration of *m* to optimise its performance in predicting subjective quality scores [6].

A classic instance of ML methods is the Feed Forward Neural Network (FFNN). The standard FFNN with one hidden layer predicts the quality of an image i_l by

$$\text{FFNN}(\mathbf{f}(i_l)) = g\Big(w_0^{(1)} + \sum_{k=1}^{K} w_k^{(1)} N_k(\mathbf{f}(i_l))\Big), \tag{2}$$

where g is the output transfer function and $N_k, k = 1, 2, \ldots, K$, are the outputs of the hidden neurons of the neural network, defined by

$$N_k(\mathbf{f}(i_l)) = h\Big(w_{k,0}^{(2)} + \sum_{j=1}^{F} w_{k,j}^{(2)} f_j(\mathbf{f}(i_l))\Big). \tag{3}$$

with $\mathbf{f}(i_l)$ defined as above and h being the hidden transfer function, which is typically a sigmoid function. Note that in their configurations neural networks do not assume features to be combined through a pre-determined model (e.g. linear combination). Neural networks are good at approximating smooth, continuous mappings of the input features [15]. Their ability to learn from complex inputs has seen them applied to a wide range of applications from automatic speaker recognition to traffic forecasting.

In this paper, the performance of a FFNN is compared to that of a more conventional metric, which assumes linear combination of the perceptually relevant features $f_j(i_l)$ (in this sense, it is not considered as a learning method, as the model m is assumed a priori, and simply tuned on data). The Principal Component Regression (PCR) is a linear regression system that combines the principal components (PCs) of the input features [16]. The variances of the selected PCs approximate the variances of the input feature values. The PCR has been successfully adopted for video objective quality assessment in [17]. However, as the PCR is a linear regression technique, it may be limited in its ability to model non-linear mapping between feature space and perceptual quality score.

3. METHOD

This study evaluates the ability of FFNNs and PCR to predict subjective image quality from SSIM features. A single feature vector for each image is extracted using the following concatenation of SSIM scalar (real-valued) features: (f_L) luminance, (f_C) contrast and (f_S) structure. In order to investigate the effect on prediction accuracy when the feature space is noisy or a sub-optimal representation, noise is added to the features, in different levels and configurations. This is expected to simulate poorly or partially informative feature spaces, for which a selection of the image information to be taken into account should be performed.

A systematic evaluation of the addition of noise to the three SSIM features was conducted. Scaled Gaussian noise was generated and added to each feature according to the following,

$$\hat{f}_j(i_l) = f_j(i_l) + \nu_j(i_l) 10^{(\Lambda/20)} \tag{4}$$

where $\nu_j(i_l)$ is sampled from a Gaussian random variable with zero mean and the same variance as the j^{th} feature: $\nu_j(i_l) \sim \mathcal{N}(0, \sigma_{f_j})$ and Λ is used to adjust the variance of the noise relative to that of the feature in dB.

The experiments were set up to gradually add noise to different (groups of) features. In the first experiment, noise was added to only one of the three features (e.g., only to luminance - f_L - leaving contrast and structure unaltered). Both PCR and FFNN were then used to predict image quality using features modified according to the setup described in be-

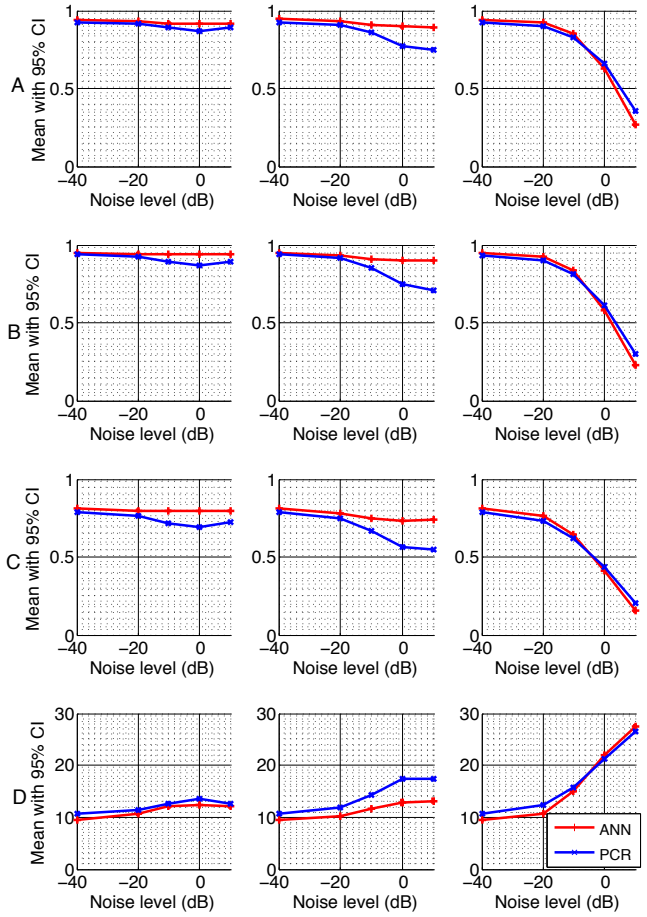

Fig. 1. Performance of FFNN and PCR, pooled according to the number of features added with noise (one, two and three, left to right). Results are (A) Pearson correlation; (B) Spearman correlation; (C) Kendall correlation and (D) RMS error.

low. The experiment was repeated three times so that every permutation, for one feature with added noise, was investigated. This was repeated for all possible numbers of noisy features (including noise-free), and for every corresponding permutation. This resulted into eight possible permutations (one experiment using noise-free features, three experiment using one noisy feature, three experiments using two noisy features and a last experiment using three noisy features). Every experiment (apart from noise-free) was repeated for five different noise levels, NL = -40, -20, -10, 0 and 10 dB. Each feature was normalised so that every input ranged between 0 and 1, consistently for training and test.

For the PCR implementation, the MATLAB functions *princomp* and *regress* were employed, and the output was normalised using a four-parameter logistic function. For the FFNN implementation, the MATLAB Neural Network Toolbox was used, configured to the Levenberg-Marquardt algorithm. It includes an early-stopping method to improve the neural network generalization performance. The hidden and output transfer functions are $h(x) = \tanh(x)$ and $g(x) = x$, respectively. The number of hidden neurons was empirically set to 3, for a total of 12 weights w to be determined in the training phase.

Subjective quality assessment databases [18, 19] are necessary for the training and validation of ML-based objective quality measures. These databases consist of distorted signals (images or videos) that are annotated with (Differential) MOS (Mean Opinion Score) values [14].

The LIVE image quality database [13] was used for this study. It contains 29 reference images and 779 distorted images, annotated with DMOS scores [20]. It includes images impaired by means of five distortion types: Gaussian blur, JPEG compression, JPEG2000 compression, white noise, and bit errors induced by a Rayleigh fading channel. It should be noticed that the LIVE database contains few reference images/videos as compared to the number of distorted images/videos. This configuration leads to the risk of the ML method being over-specialised by focusing on the few references images included in the dataset. Cross-content training and testing [8, 21] are then essential to judge the performance of ML-based quality predictors.

This experiment investigates the robustness of ML prediction accuracy due to noisy features, not training-test set size. Fixed training and test sizes were used with proportions that allowed all 3654 possible training-test combinations and corresponds to approximately ten-fold cross validation. The LIVE database was partitioned into training and test sets that contained two disjoint sets of reference images (and their distorted versions), in the proportion of 26 for the training and 3 for the test. Both the FFNN and the PCR models were evaluated on each of these training-test cases.

For both methods, recommended [14] performance indicators (Pearson, Spearman, Kendall, RMSE) were calculated for each of the 3654 training-test cases. Averages and 95% confidence limits were computed from the standard error for each performance indicator. It should be noticed that these confidence limits do not reflect the projected variance of predictive accuracy on unseen data due to pessimistic bias related to the limited dataset size [22]. However, they are still useful for the comparison of the performance of algorithms with a restricted dataset.

4. RESULTS AND DISCUSSION

Fig. 1 illustrates the difference in performance for the FFNN and PCR methods when input features are corrupted with noise. For each experiment the results are pooled according to the number of noisy features. For example; all three experiments where a single feature has noise added are treated as one (first column). Results are reported for Pearson, Spearman and Kendall correlations as well as RMS error to evaluate whether the reported trends are consistent across performance measures.

The 95% confidence intervals in Fig. 1 show that both methods exhibit a robustness across all experiments. The dif-

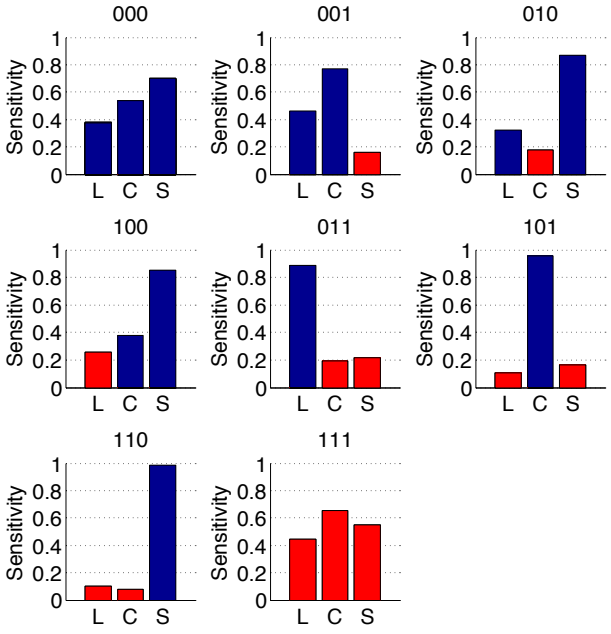

Fig. 2. Sensitivity analysis of FFNNs when noise is added with noisy features indicated by red bars. The mean normalised weight is plotted for the feature inputs: Luminance (L), Contrast (C) and Structure (S). The title of each plot is a binary mask indicating which features have noise applied.

ference in accuracy between FFNN and PCR is small when only one feature is corrupted but the FFNN shows marginal improvement for all measures. The results for two noise components are the most striking as they highlight the strength of FFNN over PCR for disregarding the noisy information in the prediction. The FFNN results remain relatively consistent across all noise levels while PCR performance decreases as the noise increases. It can be seen that there is a slight mismatch between performance indicators for FFNN as the Pearson score remains high while the RMSE is rising in extreme cases. This is caused by a sparseness of training data at high quality ratings, highlighting the need to present a range of performance indicators. When all three features are added with noise, there is little to separate FFNN from PCR.

Pooling of results allows a useful comparison of the two algorithms, but it is interesting to compare how the accuracy of the quality prediction is affected according to whether the noise is added to luminance, contrast or structure features. For the FFNN, a sensitivity analysis [23] of the trained network can shed light on this. The network sensitivity to the input of a specific feature can be measured as the magnitude of the numerical change in the network output for a given increase at that feature value. The initial value and rate of change of the input feature, as well as the values of the other features, can affect the magnitude of the output change. Therefore, for each input a range of conditions are specified and the average change in the FFNN magnitude output over all conditions is taken as the network sensitivity to that specific feature. The sensitivity of the FFNN, for feature $j = L$ (luminance), is defined as

$$s_{f_L} = \frac{1}{D(D+1)^2} \sum_{a=0}^{D-1} \sum_{b=0}^{D} \sum_{c=0}^{D} \Big| \text{FFNN}\Big(\frac{a+1}{D}, \frac{b}{D}, \frac{c}{D}\Big) - \text{FFNN}\Big(\frac{a}{D}, \frac{b}{D}, \frac{c}{D}\Big) \Big| \tag{5}$$

where $\text{FFNN}(\mathbf{f}(i_l)) = \text{FFNN}(f_L, f_C, f_S)$ is the trained neural network and $D = 10$. Similar sensitivity functions are defined for contrast (s_{f_C}) and structure (s_{f_S}). The trained networks are stored for each of the 3654 tests within each experiment. For each network a set of three sensitivities is produced, these are then normalized to the root mean square sensitivity for that network.

Fig. 2 shows the sensitivity of the FFNN to each feature (network inputs f_L,f_C and f_S) for the highest noise case (10 dB). The title of each plot is a binary mask for the three features designating which of the features has noise applied. Thus, 001 indicates that noise has been added only to feature f_S. The higher the sensitivity, the higher the relative importance of the feature in the prediction for that input configuration. In the noise-free case the structure is most important, with luminance being least important and contrast falling somewhere in the middle. When noise is added to some of the features the sensitivity of the network to the noisy features decreases significantly. The network learns that a particular feature is sub-optimal and reduces its sensitivity to it, relying on the other available features. In this sense, the FFNN seems to display intrinsic feature selection capabilities.

The sensitivity analysis allows a further observation: the network sensitivity for the SSIM components without noise (Fig. 2 with mask 000) does not match the uniform feature weighting used by SSIM as proposed by Wang et al. [11]. This confirms the assertion in Section 2 that parameter value selection in SSIM may be sub-optimal. The weights, when noise was added to all features (Fig. 2 with binary mask: 111), show that the network sensitivity has changed for contrast, but that luminance is consistent as the lowest weighted feature.

5. CONCLUSIONS

In this paper the robustness of ML-based objective image quality metrics to suboptimal feature space selection was investigated. Two techniques were compared, the first applying a PCR to pool the SSIM input features when altered with Gaussian noise, and the second using a FFNN to pool the same features. It was observed that the metric using a ML tool (the FFNN) responded in a more robust way to the addition of noise to its input features, maintaining an acceptable prediction accuracy even when a high amount of noise was added to the inputs. A sensitivity analysis revealed that this robustness could be due to the intrinsic capability of the FFNN to diminish the impact of the poorly informative (noisy) features on the final quality prediction. As a result,

the ability of FFNN to separate the wheat from the chaff in terms of input features is a key characteristic that supports the usage of ML-based tools for objective quality metrics. Future work will broaden the investigation to cover a range of datasets and other metrics besides SSIM.

6. ACKNOWLEDGEMENTS

This work was supported in part by the EC in the context of the QUALINET (COST IC 1003) project. This research was also supported by Google (AH), the Flemish Institute for the Promotion of Innovation by Science and Technology (IWT) (AB), the EPSRC (EP/J013013/1) (PK), the NWO Veni Grant 639.021.230 (JR).

REFERENCES

[1] European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), "Qualinet White Paper on Definitions of Quality of Experience," June 3, (2012).

[2] FW Campbell and JJ Kulikowski, "Orientational selectivity of the human visual system," *The Journal of physiology*, vol. 187, no. 2, pp. 437–445, 1966.

[3] JA Movshon and C Blakemore, "Orientation specificity and spatial selectivity in human vision," *Perception*, vol. 2, no. 1, pp. 53–60, 1973.

[4] SS Hemami and AR Reibman, "No-reference image and video quality estimation: Applications and human-motivated design," *Signal processing: Image communication*, vol. 25, no. 7, pp. 469–481, 2010.

[5] W Lin and C-C Jay Kuo, "Perceptual visual quality metrics: A survey," *Journal of Visual Communication and Image Representation*, vol. 22, no. 4, pp. 297–312, 2011.

[6] P Gastaldo, R Zunino, and JA Redi, "Supporting visual quality assessment with machine learning," *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, pp. 54, 2013.

[7] A Barri, A Dooms, B Jansen, and P Schelkens, "A locally adaptive system for the fusion of objective quality measures," *Image Processing, IEEE Transactions on*, vol. 23, no. 6, pp. 2446–2458, 2014.

[8] M Narwaria and W Lin, "Objective image quality assessment based on support vector regression," *Neural Networks, IEEE Transactions on*, vol. 21, no. 3, pp. 515–519, 2010.

[9] AK Moorthy and AC Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality," *Image Processing, IEEE Transactions on*, vol. 20, no. 12, pp. 3350–3364, 2011.

[10] CM Bishop, *Neural networks for pattern recognition*, Oxford university press, 1995.

[11] Z Wang, AC Bovik, HR Sheikh, and EP Simoncelli, "Image quality assessment: From error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.

[12] A Bouzerdoum, A Havstad, and A Beghdadi, "Image quality assessment using a neural network approach," in *Signal Processing and Information Technology, 2004. Proceedings of the Fourth IEEE International Symposium on*. IEEE, 2004, pp. 330–333.

[13] HR Sheikh, Z Wang, L Cormack, and AC Bovik, "Live image quality assessment database release 2," 2005, http://live.ece.utexas.edu/research/quality.

[14] Video Quality Experts Group (VQEG), "Final report from the video quality experts group on the validation of objective models of video quality assessment, Phase II," 2003.

[15] Ken-Ichi Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural networks*, vol. 2, no. 3, pp. 183–192, 1989.

[16] H Martens and M Martens, *Multivariate analysis of quality*, New York: Wiley, 2001.

[17] C Keimel, M Rothbucher, H Shen, and K Diepold, "Video is a cube," *Signal Processing Magazine, IEEE*, vol. 28, no. 6, pp. 41–49, 2011.

[18] S Winkler, "Analysis of public image and video databases for quality assessment," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 616–625, 2012.

[19] K Fliegel and C Timmerer (eds.), "WG4 databases white paper v1.5: QUALINET multimedia database enabling QoE evaluations and benchmarking," *Prague/Klagenfurt, Czech Republic/Austria*, March 2013.

[20] HR Sheikh, MF Sabir, and AC Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *Image Processing, IEEE Transactions on*, vol. 15, no. 11, pp. 3440–3451, 2006.

[21] JA Redi, P Gastaldo, I Heynderickx, and R Zunino, "Color distribution information for the reduced-reference assessment of perceived image quality," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 12, pp. 1757–1769, 2010.

[22] G Vanwinckelen and H Blockeel, "On estimating model accuracy with repeated cross-validation," in *BeneLearn 2012: Proceedings of the 21st Belgian-Dutch Conference on Machine Learning*, 2012, pp. 39–44.

[23] WS Sarle, "How to measure importance of inputs," *SAS Institute Inc., Cary, NC, USA. URL: ftp://ftp. sas. com/pub/neural/importance. html*, 2000.