Video Quality Assessment and Machine Learning: Performance and Interpretability

Søgaard, Jacob; Forchhammer, Søren; Korhonen, Jari

Published in:
Proceedings of IEEE QoMEX 2015.

Link to article, DOI:
10.1109/QoMEX.2015.7148149

Publication date:
2015

Document Version
Peer reviewed version

Video Quality Assessment and Machine Learning: Performance and Interpretability

Jacob Søgaard, Søren Forchhammer, and Jari Korhonen
Technical University of Denmark, 2800 Kgs Lyngby, Denmark

***Abstract*—In this work we compare a simple and a complex Machine Learning (ML) method used for the purpose of Video Quality Assessment (VQA). The simple ML method chosen is the Elastic Net (EN), which is a regularized linear regression model and easier to interpret. The more complex method chosen is Support Vector Regression (SVR), which has gained popularity in VQA research. Additionally, we present an ML-based feature selection method. Also, it is investigated how well the methods perform when tested on videos from other datasets. Our results show that content-independent cross-validation performance on a single dataset can be misleading and that in the case of very limited training and test data, especially in regards to different content as is the case for many video datasets, a simple ML approach is the better choice.**

I. Introduction

In video distribution and communication, a desired goal is to ensure that the video displayed to the user is of the best quality possible. An important part of achieving this goal is automatic Video Quality Assessment (VQA). One way of obtaining VQA is by building features that are relevant in regards to the subjective quality and then use a mapping between the feature space and the subjective quality space. One way of obtaining this mapping is by the use of Machine Learning (ML), which has been used in many VQA methods [1]–[11]. Despite the popularity and usual effectiveness of using ML for VQA purposes, the use of ML has several drawbacks: it is unclear how many features should be used, ML adds complexity to the method, and there is a risk of "overfitting". These drawbacks are especially prominent when ML methods resulting in complex mapping functions are used, such that the ML part of the VQA method becomes a "black box" i.e. neither intuitive nor easy to interpret. This gives rise to the trade off between performance and interpretability.

In this paper, we compare a complex ML method with an ML method that outputs a simpler mapping function. We also show how we can reduce the number of features, and we check the performance across several datasets in order to show the test performances of the methods. The interpretability [12] of the final model is related both to the number of features and to the mapping function used. To produce the feature for the VQA, we use an improved version of the No-Reference (NR) VQA method using Support Vector Regression (SVR) presented in [11] and compare it with a similar method where the ML is replaced with the Elastic Net (EN) implementation presented in this work. The method estimates parameters used during the video encoding and produces features relevant to the perceived quality. Since our focus in this work is on the ML aspects and not on the VQA method itself we only present the general outline of the method and the features produced.

VQA can be divided into the following three main categories: Full-Reference (FR), Reduced-Reference (RR) and No-Reference (NR) quality assessment. In the case of FR, the original video is accessible and the degraded version can be compared to it. In the NR scenario, the original video is not available, and therefore the quality must be estimated by solely analyzing the degraded video. The RR scenario is somewhere between these two, i.e. only some information about the original video is available for the quality estimation. For a more in-depth introduction to VQA see [13]. NR VQA methods are very useful since no additional data is transmitted along with the video signal. Thus, the algorithms can be carried out solely at the receiving end and without affecting the encoding or the amount of transmitted data.

The contributions in this paper are: the comparison of performance between two ML methods, one producing a complex mapping from the feature space to the quality score and one producing a simple mapping, the method for feature selection, and the investigation of performance inside a single dataset and across datasets. In this work, the complex method is represented by Support Vector Regression (SVR) while the simpler method is represented by Elastic Net (EN). SVR was chosen due to its popularity in objective VQA methods and EN was chosen for its general efficiency and interpretability.

The paper is organized as follows: In Section II, the background for the video features is given. In Section III, we present the ML methods and our method for feature selection. Finally, the results of our experiments are reported in Section IV along with performance and dataset evaluation.

II. Video Analysis

In this work, we use the H.264/AVC analysis method presented in [11] to define video features. Using this method, we are able to estimate important information about the video stream, such as the position of the I-frames, quantization parameters, and the PSNR. The idea of the method is to mimic the encoder by performing the intra-prediction to get an estimate of the transform coefficients without accessing the bitstream. The assumptions for the chosen methods are the following: only the decoded videos are available (i.e. the reference signal is not available and the bitstream is inaccessible), the video codec used for encoding is H.264/AVC, the QP inside a single I-frame is constant, and the GOP size is finite. The architecture of the method is illustrated in Fig. 1.

We deliberately produce a high number of features, where the inter correlation between some features might be very high, in order to check how effective our feature selection method in Section III-C is. Due to the high number of features, each feature is only described very briefly.

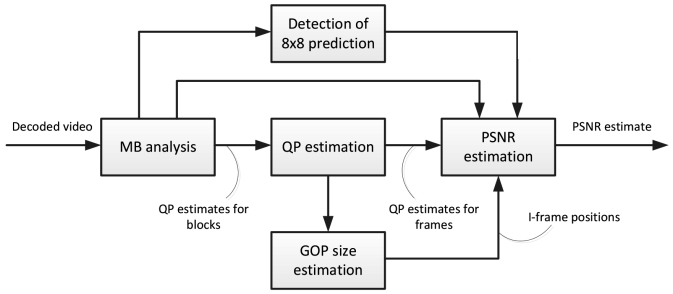

Fig. 1. Block diagram of the video analysis.

The features are based on: the estimation of the quantization parameters of all MBs in a frame with the same transform block size QP^i_{est}, an overall estimation of the frame QP value ($\tilde{QP}$), estimation of the frame PSNR $PSNR_{est}$, two measures of the reliability (or confidence) of the estimation P^i_{con} and P^i_{tot}, a weighted estimate of the frame QP denoted $w\tilde{QP}$ and temporal pooled and weighted QP estimation ($\tilde{w}QP$). Details on the features can be found in [11]. The transform block sizes are $i \in \{4, 8, 16\}$. We additionally define the averages:

$$QP_{est} = \frac{1}{3} \sum_{i=4,8,16} QP^i_{est} \quad (1)$$

$$P_{con} = \frac{1}{3} \sum_{i=4,8,16} P^i_{con} \quad (2)$$

$$P_{tot} = \frac{1}{3} \sum_{i=4,8,16} P^i_{tot} \quad (3)$$

The 44 features in Table I are based on the codec analysis from [11] and reflects the compression in the video. All of them are calculated over the I-frames of an analyzed video. We use the following notation: μ and σ are used to denote the average and the standard deviation, respectively; $\max_D$ is used to denote the maximum difference, i.e. the highest value minus the lowest value; $\max_{drop}$ is used to denote the maximum drop from one value to the next; ∇ is used to denote the gradient value of a line fitted to the values in question with the least squares criterion. When the $\max$, $\min$, and $\sum$ operator are used without indexing, indexing over the values for the I-frames is assumed. Motivated by [14], clustering is performed [15] on some information from the codec analysis values to obtain a cluster with general high values and a cluster with general low values, which is denoted by $C_h(\cdot)$ and $C_l(\cdot)$, respectively. This can be used to calculate a weighted average:

$$\mu_w(\cdot) = \frac{\sum C_l(\cdot) + w \sum C_h(\cdot)}{|C_l(\cdot)| + w\,|C_h(\cdot)|}, \quad (4)$$

where $|C_l(\cdot)|$ and $|C_h(\cdot)|$ is the number of elements in the two clusters, and the weight $w(\cdot)$ is defined by the average of the high and low cluster, denoted $\mu_H(\cdot)$ and $\mu_L(\cdot)$, respectively,

$$w(\cdot) = \left(1 - \frac{\mu_L(\cdot)}{\mu_H(\cdot)}\right)^2. \quad (5)$$

To get information about the spatial and temporal complexity in the videos we have used the spatial perceptual information measure (SI) and temporal perceptual information measure (TI) from [16] on the distorted videos. Since they are calculated on the distorted videos, they will depend on the amount of distortion. Nevertheless, the measures still contain information about the spatial and temporal complexity of the videos, which will be useful in our machine learning approach. We calculate the SI and TI values for each frame and pool the features over the frames as shown in Table II. Furthermore, we also use complexity features based on the Natural Scene Statistics (NSS) of a video. NSS have been used in several recent NR VQA methods, such as [1], [5], [6], [17]. In this work, we use NSS to calculate a measure of spatial complexity of a video and the amount of flicker. To do this, we calculate the Mean Subtracted Contrast Normalized (MSCN) coefficients for each pixel position:

$$\hat{I}(i,j) = \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + K}, \quad (6)$$

where I is the luminance of an image, (i, j) are the spatial indices of pixels, and $K = 1$ is a constant preventing numerical instabilities. $\mu(i, j)$ and $\sigma(i, j)$ are the weighted mean and standard deviation of local luminance values as detailed in e.g. [1]. For each frame we define the Spatial Complexity (SC) as the standard deviation of $\hat{I}$ (6), while the Flicker Intensity (FI) is only defined for each I-frame. To calculate this feature, the mean absolute difference between the MSCN coefficients in the I-frame and the next consecutive frame is calculated. In order to lower the dependency of the flicker metric on the content, we perform median filtering (filter length of 3) on the values and define the FI as the difference between the original values and the median filtered. The 10 features based on either SI/TI values or NSS are shown in Table II.

Due to numerical reasons, prior knowledge, e.g. the limits of QP value, is used to scale all feature values, so they are all ensured to be inside the range [-1; 1]. With this approach the scaling of the features are independent of the training, validation, and testing splits of the datasets.

TABLE I. CODEC FEATURES

#	Description
1-2)	μ and σ of QP_{est} (1)
3-7)	μ, σ, $\max_D$, ∇, and $\max_{drop}$ of QP^4_{est}
8-9)	μ and σ of P_{con} (2)
10-11)	μ and σ of P^4_{con}
12)	$\max(P_{con})$ (2)
13)	$\max(P^4_{con})$
14)	$\tilde{w}QP$
15)	$\sum \frac{QP^4_{est} P^4_{con}}{\sum P^4_{con}}$
16-20)	μ, σ, $\max_D$, ∇, and $\max_{drop}$ of wQP
21-25)	μ, σ, $\max_D$, ∇, and $\max_{drop}$ of $\tilde{QP}$
26-27)	μ of $C_h(\tilde{QP})$ and $C_l(\tilde{QP})$
28-29)	$\mu_w(\tilde{QP})$ and $w(\tilde{QP})$
30-34)	μ, σ, $\max_D$, ∇, and $\max_{drop}$ of $PSNR_{est}$
35-36)	μ of $C_h(PSNR_{est})$ and $C_l(PSNR_{est})$
37-38)	$\mu_w(PSNR_{est})$ and $w(PSNR_{est})$
39-40	μ and σ of P_{tot} (3)
41-42)	μ and σ of P^4_{tot}
43)	$\max(P_{tot})$ (3)
44)	$\max(P^4_{tot})$

TABLE II. SPATIAL/TEMPORAL COMPLEXITY FEATURES

#	Description
45-48)	μ, σ, max and min of SI
49-52)	μ, σ, max and min of TI
53)	μ of SC
54)	μ of FI

III. Machine Learning

We use the popular Support Vector Regression (SVR) method and the simpler Elastic Net (EN) method for mapping the feature values to a quality score. For the feature selection, we use a novel EN approach as detailed in Section III-C. To train, validate, and verify our system several tests as detailed in Section IV have been done.

A. Elastic Net

One of the general ML methods used to map features to a quality score is the Elastic Net (EN). The goal is to estimate the coefficients β of a regularized linear regression model:

$$\tilde{\beta} = \underset{\beta}{\operatorname{argmin}} \, ||y - X\beta||^2 + \lambda_2||\beta||^2 + \lambda_1||\beta||_1 \qquad (7)$$

where y is the target quality values, X is a feature matrix with rows of feature vectors, and λ_1 and λ_2 are regularization parameters of the $\mathbf{L}^1$-norm and the $\mathbf{L}^2$-norm, respectively. Due to the $\mathbf{L}^1$-norm in (7) the solution of an Elastic Net can generally be considered to be sparse and therefore feature selection is inherently a part of the EN method. Even so, since the SVR method is non-sparse we perform the feature selection outlined in Section III-C before training either the EN or the SVR method. Also, since our training only consist of a relatively low number of different original video contents, the additional feature selection before applying the EN could further reduce the risk of overfitting. For more information about the EN method we refer to [18]. In our experiments we use the implementation of the EN presented in [12].

The result of training an EN method, i.e. the coefficient vector $\tilde{\beta}$ (7) is simple and interpretable. The magnitude of each coefficient can be interpreted as the weight of the corresponding feature and the sign of the coefficient shows whether the feature has a negative or positive influence on the quality of the video. Also, as long as the features are scaled to the same interval, the magnitudes of the coefficients $\tilde{\beta}$ can be used to rank the importance of the features.

B. Support Vector Regression

The second ML method for mapping the features to a quality score is Support Vector Regression (SVR), specifically the ϵ-SVR method. The aim is to find a function of the feature vector $\mathbf{x}$:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x_i}, \mathbf{x}) + \beta_0, \qquad (8)$$

where α_i are the solution values, $K(\mathbf{x_i}, \mathbf{x})$ is the kernel function and β_0 is an offset. The feature vectors $\mathbf{x_i}$ where α_i is non-zero are the so-called support vectors. Interested readers are referred to [19]. We use the radial basis function as the kernel function

$$K(\mathbf{x_i}, \mathbf{x}) = e^{-\omega||\mathbf{x}-\mathbf{x_i}||^2}. \qquad (9)$$

When training the model, we search for the optimal values of three parameters, the cost and the ϵ parameter in the SVR formulation and the ω parameter in the radial basis function, in a 3-dimensional grid search. We use the implementation of ϵ-SVR as presented in [20]. Since the end result of training a SVR method, i.e. $f(\mathbf{x})$, is a sum of the outputs of the kernel function $K(\mathbf{x_i}, \mathbf{x})$ for the support vectors, it is quite hard to interpret the solution and to relate it to the individual features. We therefore consider this method to be more complex than the EN method and a "black-box" solution.

Algorithm 1 Feature Selection

while $k < K$ **do**
  Add n iid. white noise features to the feature set.
  for Fold $i = 1, \ldots, m$ **do**
    for Gridsearch over λ_1, λ_2 **do**
      Calculate $\tilde{\beta}^{i,\lambda_1,\lambda_2}_{x \in \tilde{F}_{k-1}}$ (7) with the subset of features $\tilde{F}_{k-1}$
    end for
  end for
  Set $\tilde{\beta}^i$ equal to $\tilde{\beta}^{i,\lambda_1,\lambda_2}_{\tilde{F}_{k-1}}$ with minimum cross-validation error
  Let $\tilde{\beta} = \frac{1}{m}\sum_{j=1}^{m} |\tilde{\beta}^j|$
  Update the set of chosen features: $\tilde{F}_k = \{x | \tilde{\beta}_x > \frac{1}{n}\sum_{j=1}^{n} |\tilde{\beta}_{f+j}|, x \leq f\}$
  if $\tilde{F}_k = \tilde{F}_{k-1}$ **then**
    return $\tilde{F} = \tilde{F}_k$
  end if
  $k = k + 1$
end while

C. Feature Selection

The feature selection method is inspired by [21] where n artificial noise features are introduced and used in the training of SVR with a modified kernel. In our initial experiments, we implemented this method and a similar approach using the EN method. The feature selection using EN in almost every case kept fewer features and led to better predictions than the SVR-based feature selection and we therefore only consider our EN feature selection method in the rest of this work.

The magnitude of the coefficients $|\tilde{\beta}|$ (7) in the EN can be considered as (non-normalized) weights of the features in the model. Therefore, this can be used for ranking the features and to eliminate features with less importance. Since less features means a simpler model, we would like our feature selection method to be able to trim the number of features efficiently. Also, since the amount of unique contents in video datasets often are very limited the inherent feature selection present in some methods (such as the EN) might not be enough. We therefore propose the following method for feature selection.

Let f be the number of original features (corresponding to the number of columns in the feature X as used in (7)). Denote the set of indices's of the features selected $\tilde{F} \subseteq F$, where $F = \{1, \ldots, f\}$. Use cross-validation to find the best EN performance, using grid-search to find the optimal parameters and the corresponding coefficients. The cross-validation used depends on the datasets as defined in Section IV. Let m be the number of folds in the cross-validation and K a sufficient high number of maximum iterations. Initiate Algorithm 1 with $\tilde{F}_0 = F$ and $k = 1$. The set returned gives the indices's of chosen features $\tilde{F}$.

TABLE III. EN AND SVR CROSS-VALIDATION PERFORMANCE.

	LIVE			**Lisbon**			**IVP**		
Selected features	11, **21**, 27, 45, 48, 49, 54			**4**, 7, 8, **10**, 20, **21**, **28**, 29, 33, 38, 45, 50			**4**, **10**, 13, **28**, 40, 46, 51		
Elastic Net	$\tilde{x}$	μ	σ	$\tilde{x}$	μ	σ	$\tilde{x}$	μ	σ
SROCC	**0.90**	**0.85**	**0.16**	**0.95**	**0.94**	**0.046**	0.83	0.82	**0.12**
LCC	**0.90**	**0.85**	**0.14**	0.97	**0.97**	**0.025**	0.86	0.82	**0.12**
RMSE	0.57	0.69	**0.29**	0.30	0.31	**0.068**	0.64	0.65	**0.18**
OR	**0**	**0**	**0**	0.25	0.27	**0.16**	**0.25**	0.28	**0.17**
SVR	$\tilde{x}$	μ	σ	$\tilde{x}$	μ	σ	$\tilde{x}$	μ	σ
SROCC	**0.90**	0.81	0.21	**0.95**	**0.94**	0.053	**0.88**	**0.83**	0.13
LCC	0.89	0.84	0.17	**0.98**	0.96	0.041	**0.87**	**0.83**	0.13
RMSE	**0.51**	**0.65**	0.32	**0.27**	**0.30**	0.11	**0.59**	**0.61**	0.19
OR	**0**	**0**	**0**	**0.20**	**0.23**	**0.16**	**0.25**	**0.24**	0.20

TABLE IV. EN TRAINING (GRAY CELLS) AND TEST PERFORMANCE.

Training Set	**Test Set**	**LIVE**	**Lisbon**	**IVP**
LIVE	SROCC	**0.91**	**0.90**	**0.69**
	LCC	**0.91**	**0.86**	**0.67**
	RMSE	**0.42**	**1.3**	1.5
	OR	**0**	0.80	**0.55**
Lisbon	SROCC	**0.55**	**0.95**	**0.76**
	LCC	**0.53**	**0.96**	**0.75**
	RMSE	**0.84**	**0.27**	**0.66**
	OR	**0**	**0.18**	**0.28**
IVP	SROCC	**0.43**	**0.89**	**0.89**
	LCC	**0.40**	**0.91**	**0.88**
	RMSE	1.1	1.2	0.48
	OR	**0**	0.86	**0.10**

TABLE V. SVR TRAINING (GRAY CELLS) AND TEST PERFORMANCE.

Training Set	**Test Set**	**LIVE**	**Lisbon**	**IVP**
LIVE	SROCC	0.85	0.13	0.037
	LCC	0.88	0.11	0.21
	RMSE	0.49	**1.3**	**1.2**
	OR	**0**	**0.75**	0.60
Lisbon	SROCC	0.44	0.91	0.27
	LCC	0.44	0.91	0.27
	RMSE	0.91	0.46	1.4
	OR	**0**	0.39	0.55
IVP	SROCC	0.18	0.47	0.69
	LCC	0.20	0.48	0.70
	RMSE	**0.98**	**0.88**	0.79
	OR	**0**	**0.73**	0.45

IV. RESULTS

To test the performance of our methods we have used several publicly available datasets denoted in this work as follows: LIVE [22], Lisbon [23], and IVP [24]. Videos with artifacts caused by anything other than H.264/AVC encoding were excluded from the datasets. There is no overlap in content between the datasets. The datasets have been developed using different methodologies and we therefore transform the subjective scores in order to be able to compare performance across datasets. Not all subjective data is available for each dataset, so the normal z-transform for scaling in subjective studies cannot be used. Instead we normalize the mean scores in each dataset using the mean and the variation in that dataset:

$$\tilde{MSS}_i = \frac{MSS_i - \mu_{MSS}}{\sigma_{MSS}} \tag{10}$$

where μ_{MSS} and σ_{MSS} are the mean and standard deviation of the mean subjective scores in the dataset. The scores in the dataset MSS_i are either MOS or DMOS. Note, that even after this scaling there is a lot of uncertainty when testing performance across datasets, due to the differences in methodology, content, resolution, encoder setup, and even cultural differences. Therefore, we mostly concentrate on the correlation results across datasets, since they do not depend on shifting and scaling of the scores.

The overall testing procedure used in this work is to use feature selection on a training set using cross-validation, find optimal parameters for the ML methods using the training set and cross-validation, train the ML method on the whole training set, and finally use this model on an independent test set. We use the following measures to report the performance: the Spearman Rank Order Correlation Coefficient (SROCC), the Linear Correlation Coefficient (LCC), the Root Mean Square Error (RMSE), and the Outlier Ratio (OR) [25]. In the case of cross-validation, we report the median $\tilde{x}$, mean μ, and standard deviation σ of these measures, which reflects the average performance and the robustness.

The features selected (out of the sets in Tables I-II) using different training sets are listed at the top of Tables III and VI. The performance using cross-validation on the training set can also be seen in these tables for the EN and SVR method. The training and test performance of the final models using different training sets can be seen in Tables IV and VII for the EN method and in Tables V and VIII for the SVR method. The best score between the two ML methods in the different scenarios are highlighted in bold in all tables. In the selected feature row in Table III and VI, the features chosen that overlap with chosen features from other training sets of the same size has been highlighted. For simplicity, all performance measures lower than 10^{-10} has been round to 0.

Due to the varying number of unique content in the training sets, the cross-validation is performed in slightly different ways, but always using content-independent splits, e.g. *all* videos that have been coded using the same original video is either in the training or in the validation set of a cross-validation fold and never split in any way between the two sets. When only using 1 dataset as training, the cross-validation was performed by leaving out 2 contents out of the total 10-12 contents for validation in each cross-validation fold. In this case, the experiments were done for all possible content-independent splits between training and validation data. This procedure results in $\binom{n}{k}$ splits, where $k = 2$ and n is the total number of videos. When the training set consisted of two or more datasets, standard 10-fold cross-validation was used with

TABLE VI. EN AND SVR CROSS-VALIDATION PERFORMANCE ON TWO DATASETS.

Selected features	**LIVE - Lisbon** **4**, **10**, **21**			**IVP - Lisbon** **4**, 7, **10**, **21**, 26, 28			**LIVE - IVP** **21**		
Elastic Net	$\tilde{x}$	μ	σ	$\tilde{x}$	μ	σ	$\tilde{x}$	μ	σ
SROCC	**0.80**	**0.76**	0.088	**0.85**	**0.84**	0.036	**0.70**	**0.70**	0.022
LCC	**0.79**	**0.75**	0.079	**0.84**	**0.84**	0.034	**0.65**	**0.65**	0.017
RMSE	**0.76**	**0.78**	0.11	**0.62**	**0.63**	0.12	**0.84**	0.94	0.25
OR	**0.29**	**0.31**	0.078	0.53	0.55	0.12	0.23	0.24	0.083
SVR	$\tilde{x}$	μ	σ	$\tilde{x}$	μ	σ	$\tilde{x}$	μ	σ
SROCC	0.55	0.55	**0**	0.78	0.78	**0**	0.43	0.43	**0**
LCC	0.33	0.33	**0**	0.63	0.63	**0**	0.56	0.56	**0**
RMSE	1.1	1.1	**0**	0.84	0.84	**0**	0.92	**0.92**	**0**
OR	0.34	0.34	**0**	**0.51**	**0.51**	**0**	**0.19**	**0.19**	**0**

TABLE VII. EN TRAINING (GRAY CELLS) ON TWO DATASETS AND TEST PERFORMANCE ON INDEPENDENT DATASET. TRAINING PERFORMANCE IS MEASURED ON BOTH DATASETS TOGETHER.

Training Set	**Test Set**	**LIVE**	**Lisbon**	**IVP**
LIVE - Lisbon	SROCC	0.82	0.82	**0.74**
	LCC	0.81	0.81	**0.74**
	RMSE	0.74	0.74	0.77
	OR	0.34	0.34	**0.30**
IVP - Lisbon	SROCC	**0.56**	0.89	0.89
	LCC	**0.56**	0.89	0.89
	RMSE	**0.83**	0.49	0.49
	OR	**0**	0.55	0.55
LIVE - IVP	SROCC	**0.70**	**0.90**	**0.70**
	LCC	0.65	0.87	0.65
	RMSE	0.75	**0.58**	0.75
	OR	**0.17**	**0.54**	**0.17**

TABLE VIII. SVR TRAINING (GRAY CELLS) ON TWO DATASETS AND TEST PERFORMANCE ON INDEPENDENT DATASET. TRAINING PERFORMANCE IS MEASURED ON BOTH DATASETS TOGETHER.

Training Set	**Test Set**	**LIVE**	**Lisbon**	**IVP**
LIVE - Lisbon	SROCC	**0.84**	**0.84**	0.65
	LCC	**0.86**	**0.86**	0.71
	RMSE	**0.51**	**0.51**	**0.73**
	OR	**0.23**	**0.23**	0.35
IVP - Lisbon	SROCC	0.49	**0.91**	**0.91**
	LCC	0.49	**0.92**	**0.92**
	RMSE	0.87	**0.40**	**0.40**
	OR	**0**	**0.16**	**0.16**
LIVE - IVP	SROCC	0.66	0.87	0.66
	LCC	**0.71**	**0.88**	**0.71**
	RMSE	**0.72**	0.66	**0.72**
	OR	0.20	0.68	0.20

the additional restriction imposed by content-independency. In this case, 2-3 contents is used for validation in each fold.

An interesting comparison is to look at the average cross-validation performance against the training. Generally, one would expect the average cross-validation performance to be lower than the training performance of the model trained on all the training data with the same parameters. As can be seen when comparing Table III ($\tilde{x}$ columns) and Table IV (gray cells), this is true for most performance measures for the EN method when using 1 dataset as training (only for LCC the correlation is 0.01 worse in training performance for Lisbon). However, for SVR this seems not to be the case. Comparing Table III ($\tilde{x}$ columns) and Table V (gray cells), the worst performances for training compared to average cross-validation performance for SVR are: SROCC (0.19), LCC (0.17), RMSE (0.20), and OR (0.20) for IVP, where the differences has been noted in parentheses. This suggests that the cross-validation performance for SVR in this case is not a good indicator of the actual performance of the model. This might be due to the low number of training samples (only 8-10 contents) and validation samples (2 contents). Doing the same comparison for training on two datasets, Tables VI and VII, and Tables VI and VIII, the conclusion is much the same for EN (the only worse training performance is LIVE-Lisbon OR (0.03)), but for SVR the training performance is also better in all cases.

When comparing the average cross-validation performance against the test performance on independent datasets with training on a single dataset, it is evident that especially for SVR the cross-validation performance is not a very good indicator of the actual performance. In all cases the correlation coefficients on the independent datasets are very poor (below 0.5). The EN method has better performance with only one test performance with correlations below 0.5 (IVP training, LIVE testing) while test cases has correlations above 0.75 and the best correlation as high as 0.91. Using two datasets improves the test performance for SVR significantly on the IVP and Lisbon dataset, while the performance is still lacking on LIVE. For the EN method only the RMSE and OR performance seems to benefit from the larger training set, which especially improves on the Lisbon dataset. This seems to indicate that the more complex method, SVR, benefits more from being trained on more data.

The amount of selected features in the different training cases ranges from 1 to 12 features. The amount of chosen features that do not overlap across datasets are much higher for single training sets (Table III) than for two training sets (Table VI). This is most likely due to the fact, that inside a single dataset some features can be very relevant for the quality of the videos, but due to the differences in the datasets they might not be as relevant in other datasets. Thus, when training on two datasets, the features selected are more robust. The features that are most commonly selected are 4, 10, 21, and 28, which are related to different temporal poolings of the estimated QP value and the confidence of the estimation. In [11] the same VQA method is also used to produce 15 features and SVR is used to map it to quality scores with the following reported means of content-independent cross-validation: $SROCC = 0.88, LCC = 0.86$ for LIVE and $SROCC = 0.95, LCC = 0.95$ for Lisbon. In both cases we get better or equal performance with a lower number of features using either the EN or the SVR method.

When comparing the EN and SVR method, the simpler EN

method seems to outperform the more complex SVR method in most cases, especially with regards to test performance on independent test sets in Tables IV-V, and VII-VIII. Even in the few cases where SVR is slightly better than EN, it is still preferable to use the EN method due to its simplicity and the interpretability of the model. An example of a final EN-model is for training on the LIVE-Lisbon dataset, where the features 4, 10, and 21 get the corresponding model β coefficients -0.63, 0, and -2.4, meaning that feature 10 (the mean of the reliability measure P^4_{con}) is discarded for the final EN model and the quality prediction is a weighted sum of standard deviation of the estimated QP value for prediction blocks of size 4 (feature 4) and the average estimated frame $\tilde{QP}$ (feature 21), both having a negative impact on the quality, and the last feature being most important. Since the EN model contains regularization, future work could include investigation of the amount of influence the regularization part has on the generally superior performance of EN compared to SVR.

From the results it is also evident that the videos in the LIVE dataset are generally difficult to rank (best test correlation equal to 0.56), but easy to predict inside the 95% confidence interval used for the OR (OR is equal to 0 in every test case). The Lisbon videos on the other hand seems easy to rank (best test correlation equal to 0.91), but difficult to predict accurately (lowest test OR equal to 0.54). IVP dataset seems somewhat between the two other datasets with regards to performance (best test correlation equal 0.76 and best OR equal to 0.28). This is somewhat to be expected, since range of compression levels in the LIVE dataset is smaller than the Lisbon dataset, and the IVP dataset contains some untraditional content, such as animated video.

Our results also show that it can be misguiding to measure and report the performance of a VQA method only on a single dataset with limited number of original contents, even if it is done using content-independent cross-validation. Even if several independent datasets are used to report the performance of a method, but the method is still trained, validated, and tested on each dataset separately, one might get unrealistic high performance.

V. Conclusion

In this paper, a feature selection method and two ML methods for VQA were presented and compared. We achieve good VQA performance (up to a SROCC of 0.87 when testing EN on an independent dataset) with a relatively low number of features (from 1 to 12). We have shown, that in the case of VQA with limited training data, the simple and easy to interpret ML method EN can perform just as well as the more complex SVR method, and even outperforms the SVR method in some cases. Furthermore, it has been shown how the content-independent cross-validation performance on datasets with limited content can be misleading compared to even the training performance over all the content in the same dataset.

References

[1] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695–4708, 2012.

[2] P. Gastaldo, S. Rovetta, and R. Zunino, "Objective assessment of MPEG-video quality: a neural-network approach," in *Proc. Int'l. Joint Conf. Neural Networks*, vol. 2, 2001, pp. 1432–1437.

[3] S. Mohamed and G. Rubino, "A study of real-time packet video quality using random neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1071–1083, 2002.

[4] P. Le Callet, C. Viard-Gaudin, and D. Barba, "A convolutional neural network approach for objective video quality assessment," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1316–1327, 2006.

[5] K. Zhu, K. Hirakawa, V. Asari, and D. Saupe, "A no-reference video quality assessment based on laplacian pyramids," in *Proc. IEEE Int'l Conf. Image Process.*, Melbourne, 2013, pp. 49 – 53.

[6] M. A. Saad, A. C. Bovik, and C. Charrier, "Blind prediction of natural video quality," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1352–1365, 2014.

[7] M. Narwaria, W. Lin, and A. Liu, "Low-complexity video quality assessment using temporal quality variations," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 525–535, 2012.

[8] S. Argyropoulos, A. Raake, M.-N. Garcia, and P. List, "No-reference video quality assessment for SD and HD H.264/AVC sequences based on continuous estimates of packet loss visibility," in *Proc. Third Int'l Workshop Quality Multimedia Experience*, 2011, pp. 31 – 36.

[9] N. Staelens et al., "Constructing a no-reference H.264/AVC bitstream-based video quality metric using genetic programming-based symbolic regression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 8, pp. 1322–1333, 2013.

[10] M. Narwaria and W. Lin, "SVD-based quality metric for image and video using machine learning," *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, no. 2, pp. 347–364, 2012.

[11] J. Søgaard, S. Forchhammer, and J. Korhonen, "No-reference video quality assessment using codec analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. PP, no. 99, 2015.

[12] K. Sjöstrand, L. H. Clemmensen, R. Larsen, and B. Ersbøll, "SpaSM: A matlab toolbox for sparse statistical modeling," *Journal of Stat. Software*, 2012.

[13] J. G. Apostolopoulos and A. R. Reibman, "The challenge of estimating video quality in video communication applications," *IEEE Signal Process. Mag.*, pp. 156–160, May 2012.

[14] J. Park, K. Seshadrinathan, S. Lee, and A. C. Bovik, "Video quality pooling adaptive to perceptual distortion severity," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 610–620, 2013.

[15] M. Ankerst, M. M. Breunig, H. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure." ACM Press, 1999, pp. 49–60.

[16] *Recommendation ITU-T P.910: Subjective video quality assessment methods for multimedia applications*, Int'l Telecom. Union Std., 2008.

[17] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a 'completely blind' image quality analyzer," *IEEE Signal Processing Letters*, 2012.

[18] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Stat. Society: Series B*, vol. 67, no. 2, pp. 301–320, 2005.

[19] S. R. Gunn, "Support vector machines for classification and regression," University of Southampton, School of Electronics and Computer Science, ISIS technical report, May 1998.

[20] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intelligent Systems Technology*, vol. 2, pp. 27:1–27:27, 2011, http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[21] J. Bi, K. Bennett, M. Embrechts, C. Breneman, and M. Song, "Dimensionality reduction via sparse support vector machines," *The Journal of Machine Learning Research*, vol. 3, pp. 1229–1243, 2003.

[22] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video," *IEEE Trans. Image Process.*, vol. 19, pp. 1427–1441, 2010.

[23] T. Brandão and M. P. Queluz, "No-reference quality assessment of H.264/AVC encoded video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1437–1447, 2010.

[24] S. Li, L. Ma, and K. N. Ngan, "Full-reference video quality assessment by decoupling detail losses and additive impairments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 7, pp. 1100–1112, 2012.

[25] *Recommendation ITU-T P.1401: Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models*, Int'l Telecom. Union Std., 2012.