Predictive No-Reference Assessment of Video Quality

Maria Torres Vega[a], Decebal Constantin Mocanu[a], Stavros Stavrou[b], Antonio Liotta[a]

[a] *Department of Electrical Engineering, Eindhoven University of Technology Eindhoven, the Netherlands*
[b] *Faculty of pure and Applied Sciences, Open University of Cyprus Nicosia, Cyprus*

Abstract

Among the various means to evaluate the quality of video streams, light-weight No-Reference (NR) methods have low computation and may be executed on thin clients. Thus, these methods would be perfect candidates in cases of real-time quality assessment, automated quality control and in adaptive mobile streaming. Yet, existing real-time, NR approaches are not typically designed to tackle network distorted streams, thus performing poorly when compared to Full-Reference (FR) algorithms. In this work, we present a generic NR method whereby machine learning (ML) may be used to construct a quality metric trained on simplistic NR metrics. Testing our method on nine, representative ML algorithms allows us to show the generality of our approach, whilst finding the best-performing algorithms. We use an extensive video dataset (960 video samples), generated under a variety of lossy network conditions, thus verifying that our NR metric remains accurate under realistic streaming scenarios. In this way, we achieve a quality index that is comparably as computationally efficient as typical NR metrics and as accurate as the FR algorithm Video Quality Metric (97% correlation).

Keywords: Quality of Experience, No-Reference Video Quality Assessment, Supervised Machine Learning

1. Introduction

Low complexity No-Reference (NR) video quality methods have the potential to provide real-time video quality assessment and automated quality control, for instance in the context of video streaming on demand [1], peer to peer services [2, 3] or real-time network management [4, 5]. This is because simple NR algorithms are computationally light and do not require comparing the video stream under scrutiny with its original (unimpaired) benchmark, as would be the case of Full-Reference (FR) methods [6].

Email address: m.torres.vega@tue.nl (Maria Torres Vega)

Due to their particular methodology, computational requirements and functional limitations, neither FR methods nor subjective evaluations are viable to automate quality control processes, whereby both scalability and speed are required. Subjective studies are performed offline but are instrumental in understanding quality perception, i.e. Quality of Experience (QoE) [7, 8]. On the other hand, FR algorithms such as the Video Quality Metric (VQM) [9] have proven to correlate well with the human vision system [10, 11] and this is the reason why many studies use them to benchmark other simpler algorithms, rather than being used directly in video management applications [12, 13].

This is in fact the approach we use in our work, where we aim to introduce an NR method that combines the efficiency (and applicability) of simple NR metrics with the accuracy that is typically achieved only through heavyweight FR methods. In this way, our method enables a whole new range of applications, such as real-time assessment of video-on-demand services or network provider's quality management. This cannot be achieved through typical, low complexity NR metrics, since these are not designed to tackle network-distorted streams, thus performing poorly when compared to Full-Reference (FR) algorithms [14] [8]. On the other hand, FR methods are functionally inapplicable in real-time streaming scenarios, whereby both the original and the distorted streams are required. Subjective assessment is impractical too, due to the large scope of testing conditions that ought to be presented to the subjects.

Nowadays, Video Quality Assessment (VQA) methods and metrics are drawn from knowledge in human QoE and perception [15]. At its essence, VQA is a subjective matter, best judged by human subjects, as in subjective studies and subjective analyses [16]. Typically, sample people (chosen from different representative categories) rate video quality (or quality variations), under controlled conditions, following well-established methods [17]. The outcomes are given in terms of Mean Opinion Score (MOS) or any other derived metric. Although well-aligned to human perception, subjective studies are costly, time-consuming and prone to human bias. They are fundamental to the various applications of VQA, yet great effort has been directed towards mimicking subjective studies through completely automated processes and algorithms, as in objective QoE [18].

Traditionally, objective methods use as input the original reference signal (e.g. image, video, audio) and a distorted version. In our context, this will be a video sequence distorted by compression and network impairments. FR QoE aims to estimate the perceptual degradation in the distorted sequence, compared to the reference sequence [19, 10]. Perhaps the simplest, most popular and less accurate among FR algorithms is the Peak Signal to Noise Ratio (PSNR) [20], derived directly from the Mean Square Error [21]. A better compromise between complexity and accuracy is offered by the Structural Similarity (SSIM) [22, 23], which combines video luminance, contrast and structure to evaluate the quality degradation at frame-by-frame level. When the inter-frame degradations are of interest (for instance in the presence of network-impaired video streams), VQM is a better option [10].

Although not perfectly, FR metrics provide the best correlation with human

perception, but are not always applicable in real systems due to the requirements to have both the reference and the distorted sequence available. Also the more accurate FR metrics are computationally demanding and are, instead, more effective to generate offline benchmarking, as we do in our study.

To the other end of the spectrum, stand the NR metrics [15], that operate merely on the distorted sequence (e.g. the video stream rendered after network transmission, as in our case) and measurements from the network. These metrics usually focus on specific features [24, 25, 26], which are only indicative of quality and do not always correlate well with subjective or FR results [7]. In previous research, we analyzed a range of state-of-the-art NR features (computable in real-time) on a large video dataset and involving packet losses in the 0-10% range [14]. We showed how different metrics capture diverse types of distortions, concluding that none of the analyzed low complexity features is universally effective (they are accurate only under limited operational conditions). Also, all metrics failed under lossy networks.

Given the complexity of FR methods and the inaccuracy of low complexity NR methods, the aim of this paper is to explore how Machine Learning (ML) may lead to an accurate NR method, without increasing the complexity of the assessment process. This direction is currently being explored in the development of NR algorithms. Promising examples are the Adaboost approach for assessing artifacts levels in videos, by Vink et al. [27]; the bitstream based artificial neural network, by Shahid et al. [28]; the artificial neural network for jerkiness evaluation, by Xue et al. [29]; and the regression framework for estimating the objective quality index (SSIM or PSNR), by Shanableh [30].

A key limitation of current studies is that they have focused on video distortions generated either by compression or by synthetic impairments [31] [28]. Per contra, the assessment of realistic streaming scenarios that we are scrutinizing involves large datasets of videos distorted through a representative range of network impairments. Yang et al. have provided an early study of network-impaired videos based on a small dataset [32]. Yet our aim is to introduce a method that is proved to work on the breath of conditions faced by network and service providers, who are nowadays required to manage QoE in real-time. Our method analyses the received video stream in terms of eight NR features (both on the bitstream and the pixel levels) in addition to sensing the network to obtain two network measurements (nominal bitrate and estimated level of packet loss) in real-time. These ten features serve as input to a Supervised Learning (SL) algorithm that, based on previously learned samples of video quality through an offline training process on the server side, performs a predictive NR assessment of the quality of the video stream under scrutiny.

We extensively tested our method in a large video dataset that we generated starting from ten video types of the Live Video Database [33]. We then enhanced the dataset, generating new network impaired videos, for a total of 960 samples (Table 1) as detailed in [14]. Herein, we present the method and its evaluation, finding an overall correlation to VQM higher than 97%. Testing out this method on nine representative ML algorithms allowed us to show the generality of our approach, whilst finding the best performing algorithms.

Table 1: Video dataset parameters range in terms of video types (acronym, name and description), compression and network packet loss ratio (960 samples in total).

Video type			Compression	Packet loss
Acronym	Name	Description	64kbps	PL0%
bs1	Blue Sky	Circular motion; Blue sky and trees	640kbps	PL0.5%
mc1	Mobile Calendar	Pan, horizontal tor train; Calendar vertical	768kbps	PL1%
pa1	Pedestrian Area	Still; People on intersection	1024kbps	PL1.5%
pr1	Park Run	Pan; Person across a park	2048kbps	PL2%
rb1	River Bed	Still; River bed, pebbles in the water	3072kbps	PL2.5%
rh1	Rush Hour	Still; Rush hour traffic on the street	4096kbps	PL3%
sf1	Sunflower	Still; Bee over sunflower	5120kbps	PL3.5%
sh1	Shields	Pan, still, zoom; Person across display		PL4%
st1	Station	Still; Railway track, train and people		PL4.5%
tr1	Tractor	Pan; Tractor across the fields		PL5%
				PL10%

Table 2: PCC correlations to VQM of the eight NR metrics and SSIM. Cell colors give qualitative correlation levels: green (best), yellow (median), and red (worst).

V.T.	CX	MO	NM	NR	BM	BR	BL	JE	SSIM
bs1	0.168	0.011	-0.488	0.118	-0.013	-0.637	0.439	-0.701	0.735
mc1	0.663	0.0177	-0.644	0.538	-0.065	-0.818	0.085	0.368	0.903
pa1	0.291	-0.028	0.646	0.457	0.11	0.465	0.442	0.057	0.883
pr1	0.304	-0.164	-0.704	-0.122	0.01	-0.2	0.49	0.607	0.688
rb1	0.533	0.57	0.432	0.514	0.546	0.44	0.2	-0.59	0.2555
rh1	0.391	-0.475	0.1655	0.32	0.351	0.369	-0.686	-0.671	0.91
sf1	0.413	-0.4141	-0.728	0.136	0.5162	0.42	0.552	-0.415	0.84
sh1	0.413	-0.0925	-0.352	0.468	0.216	-0.72	0.47	0.53	0.87
st1	0.47	-0.33	-0.65	0.35	0.4372	-0.21	0.6322	-0.267	0.7554
tr1	0.53	-0.178	0.087	0.738	0.51	0.157	0.307	0.581	0.885
All	0.418	-0.108	-0.2233	0.352	0.262	-0.073	0.294	-0.05	0.7725
	±0.134	±0.28	±0.49	±0.237	±0.2275	±0.487	±0.362	±0.514	±0.187

The remainder of this paper is organized as follows. Section 2, provides a state of the problem at hand, summarizing our earlier study of NR metrics. In Section 3, the proposed predictive NR method is presented. The evaluation methodology is described in Section 4. Our findings are discussed in Sections 5 to 8, in relation to different test cases. The state-of-the-art on NR metrics in general and the use of ML techniques in particular, is given in Section 9. Finally, Section 10 draws conclusions, highlighting our key contributions.

2. Previous work

The experimental survey we presented in [14] served as motivation and starting point for this work. Our purpose was to study the performance of low complexity NR metrics in the assessment of network-impaired video quality and, if possible, to pinpoint NR features which could serve as alternative to FR metrics in situations with thin clients (such as mobile devices) or where real-time quality assessment is required in real-world streaming scenarios. These involve network-impaired video streams, which are rather different from synthetically-impaired streams [8, 34]. We studied eight well-known NR metrics, over a wide range of

video types, compressions and lossy network conditions, benchmarking the NR accuracy against the FR metric VQM. We concluded that none of the NR metrics was able to perform an accurate assessment on a general base, i.e. over all video types, compressions and network conditions. In that way, none of the simple NR metrics under scrutiny could serve as alternative to the highly complex FR methods. Most importantly, all metrics failed under lossy networks. However, it also emerged, that each metric exhibited specific operational boundaries, within which the performance was accurate to the benchmark.

Armed with these results, our next research hypothesis was that it would theoretically be possible to derive a hybrid NR metric characterized by a much broader operational boundary. However, before we introduce this new metric (Section 3), it will help to summarize the key methodology and findings which we have further detailed in [14]. We studied eight NR features, namely complexity (CX), motion (MO), blockiness (BL), jerkiness (JE), average blur (BM), blur ratio (BR), average noise (NM) and noise ratio (NR). We also included SSIM, a well-known FR algorithm, which is less accurate and complex than the VQM benchmark [9]. All these metrics were evaluated over a range of $0-10\%$ packet loss rates, what is considered to be one of the most critical types of network impairments [8, 34]. The other parameters were video type and bitrate.

The ten original raw, 10 seconds, 25fps video types were obtained from the Live Quality Video Database [33]. We compressed them at eight levels using MPEG4 part 10/H.264 and a resolution of 768x432. The selection of the encoding bitrates has been done in a way as to obtain the most diverse variety of video qualities. For example, very low quality transmissions (64kbps) are nowadays, with the currently used systems and Internet speeds, highly unlikely to occur. Each sample was then impaired at twelve packet loss rates, obtaining 960 videos as specified in Table 1.[1]

Next, we carried out a detailed evaluation of the whole dataset, according to all the NR metrics under scrutiny, including also the FR metric SSIM. Blur, noise, blockiness, ringing or temporal impairments have been quantified for measuring the end-user's quality [15]. Thus, for our study to be as broad as possible we selected light-weight metrics with demonstrated correlation to the human vision system. From all possible low-complexity metrics, we selected eight representatives, six on the pixel layer (blur and ratio of noise and blur, blockiness and jerkiness) and two on the bitstream layer (complexity and motion). These eight metrics were benchmarked against VQM through the Pearson correlation coefficient (PCC) [35]. The key results are summarized in Table 2. While rows one to ten of the table show the results for each of the specific video types, in row eleven the overall averaged and deviation correlation values can be seen.

Looking at the overall correlations (last row), we observe that none of the

[1] Upon acceptance of this paper, we shall release the whole dataset and software implementation at www.tue.nl/universiteit/faculteiten/electrical-engineering/onderzoek/onderzoeksgroepen/electro-optical-communications-eco/research/network-management-and-control/datasets/network-impaired-video-dataset/.

NR metrics achieves an acceptable correlation (50%). The best NR performant is complexity (CX), with roughly an average correlation of 42%. Noise ratio (NR), blockiness (BL) and the average blur (BM) reach roughly 30%, while the average noise (NM) anti-correlates to the benchmark. Also, the standard deviations are noticeably high in all cases, which denotes a broad performance variation across the video dataset. This can be seen directly by looking at the spread of the cell values and colors.

As expected, being an FR metric, SSIM gives much better performance than any of the NR ones (rightmost column), with an overall correlation to VQM of about 77%. Yet the standard deviation is still relatively high, indicating that SSIM too will have a limited operational boundary. Further evaluations unveiled that in fact SSIM starts failing at higher packet losses (between 1.5% and 4%) and by various degrees, depending on the video type [14].

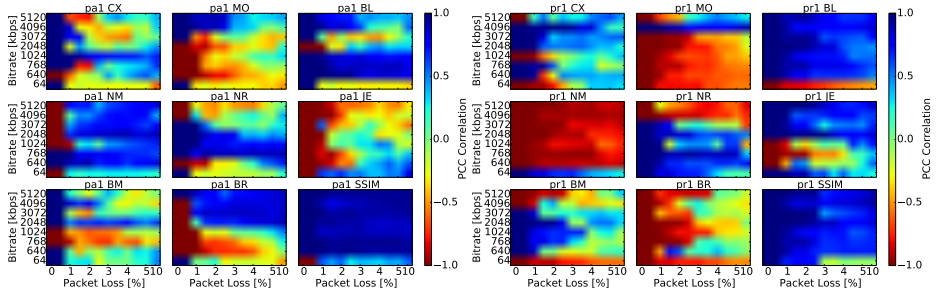
In order to narrow down the working limits of the various NR metrics, in [14] we went on analyzing the different video types individually (Figure 1), with particular attention to compression level (Y axes) and packet loss (X axes). In Figure 1, maximum correlation to VQM is shown in dark blue, while maximum anti-correlation is in dark red. Again we see that, although the analysis has been narrowed down (instead of being averaged across the whole dataset), none of the metrics operates accurately beyond some fairly narrow conditions.

It is encouraging, though, that specific blue (well correlated) areas emerge for all the NR metrics under scrutiny. For example, in the pedestrian area video (pa1, Figure 1a) blockiness performs well at low bitrates and on a broad range of packet loss. In the park run video (pr1, Figure 1b), the noise ratio performs well on medium to low bitrates, but only when packet loss is low. At the same time, jerkiness offers good complementary conditions (high bitrate, broad range of packet loss). These results encouraged us to pursue the study of hybrid metrics that would combine the strengths of individual metrics, as explained in the remainder.

3. Predictive NR Video Quality Method

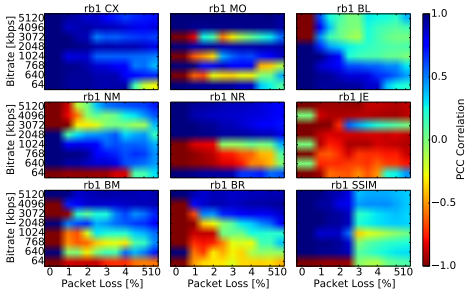
In this section, we present our predictive NR video quality method. Figure 2 shows the block diagrams for the processes running, respectively, on the server side and in the clients.

As with any prediction-based method, the accuracy of the model will substantially depend on the characteristics of the dataset used for training. In the case of our video service, the training set is composed by a number of video type samples stored in the server (further details about the training set and process can be found in Section 4.2). Each sample in the training set includes the eight NR features of Table 2 (both in the pixel and the bitstream layers), two network condition parameters (packet loss rate and bitrate) and the ground truth quality index. This training set is used (in the server) to maintain the quality prediction function, which is then employed on the client side to compute our predictive NR video quality assessment metric.



(a) Correlation maps for video pa1.

(b) Correlation maps for video pr1.



(c) Correlation maps for video rb1.

Figure 1: Pearson correlation to VQM of the eight NR metrics (CX, MO, NM, NR, BM, BR, BL, JE) and the SSIM FR metric, considering bitrates between 64 and 5,120 Kbps and packet losses between 0 and 10%. Video types: a) Pedestrian Area (pa1); b) Park run (pr1); and c) River bed (rb1). The original (unimpaired) videos were obtained from the Live Quality Video Database [33]. Network impairments were incurred by streaming videos through the PacketStorm network emulator [36].

At service launch, the service provider will already have a representative video types set (e.g., sport, action movies, cartoons, and so forth); thus an initial prediction model can be constructed (and made available to the client side). When a completely new video type is added or a completely different condition is detected, the prediction model will be less accurate. Yet, over the time the model will be updated based on new types and conditions (by means of feedback loops from the clients) and, what is more important, the chances of getting new video types and conditions will rapidly diminish. In this way, the server runs a process in the background in which the SL model is trained with the available video samples and new models ($\hat{f}_{server}$) are uploaded to the clients (on a continuous or periodic basis).

On the other end of the transmission link, the video client employs the SL model trained by the server, to generate its prediction-based quality metric (Q_p). During a streaming session, the client characterizes the incoming video in terms of NR features and real-time network conditions, matching this information against the prediction model to generate the NR quality index. Given the fact that the process in the server is executed as an independent (background) routine, the real-time quality assessment algorithm is not tied to it. The two

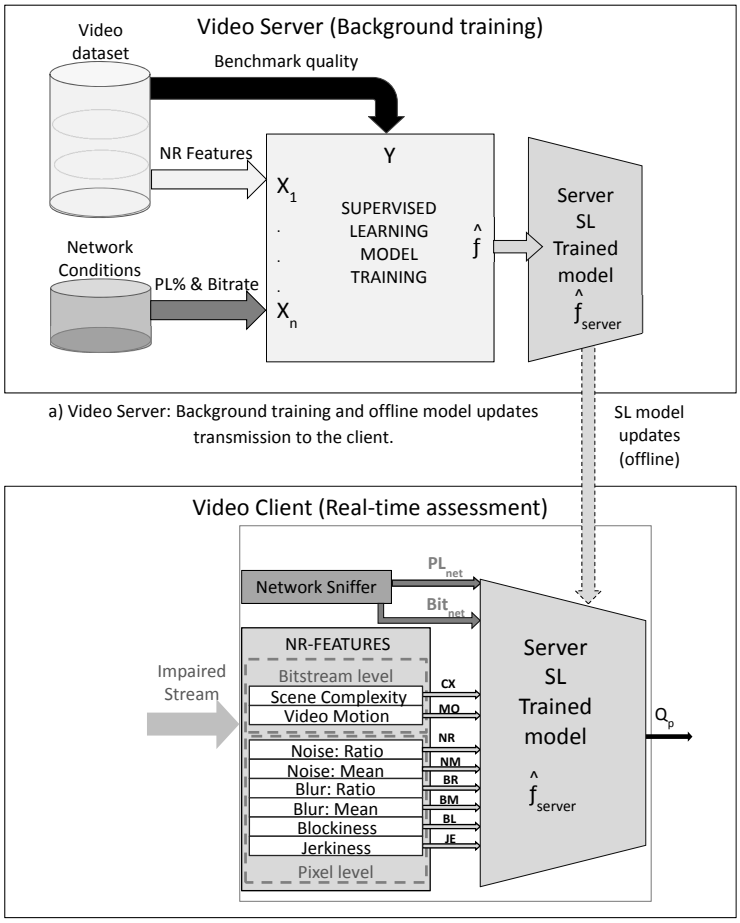


Figure 2: Block diagram of the predictive NR video quality assessment method. a) server side (background training); b) client side (real-time assessment).

processes, in client and server, proceed independently from each other, the model is not updated from server to client in an online manner; hence, our method falls within the NR category. In principle, our model could also be implemented as an RR metric, whereby the model parameters are passed to the client online. Yet, this is not the scenario we have explored in this manuscript.

Selecting the features that better characterize the video streams, are effective in the SL training process (in the server) and, ultimately, generate an accurate quality metric (in the clients), is not trivial. Our choice was driven by a preliminary (extensive) evaluation of classic, low computation and real-time NR video features (Section 2 [14]), where we studied their operational boundaries. From our accuracy study, as shown in the previous section, we saw that while none of the metrics under scrutiny could provide an overall good performance, all of

them had their own working boundaries. We then followed the intuition that, if individual metrics would work accurately under specific conditions, a functional combination of those metrics could work on a broader range of conditions.

Bitstream parameters have been shown to provide good results when assessing networked videos [13, 37]. However, the video characteristics and content will have a substantial influence on the assessed quality, thus, our selection and prior study focused not only on bitstream features but on low complexity pixel level metrics that can be obtained in real time from the received frames.

In general, a video stream can be characterized by several parameters, i.e. the ones that would allow differentiating among different video types. Parameters regarding the video scene composition have been demonstrated to affect quality to a large extent [38]. Among these, scene complexity and video motion have proven to correlate well with video quality [39]. Scene visual complexity is conventionally defined as the level of detail or intricacy contained within an image or frame [40] or the number of objects or elements present in the frame, whereas video motion is the amount of movement in the video [39]. Both features can be empirically obtained from the codec using the equations shown below (Equation 1) [38].

$$C = \frac{Bits_I}{2 * 10^6 * 0.91^{QP_I}} \qquad M = \frac{Bits_P}{2 * 10^6 * 0.87^{QP_P}} \tag{1}$$

Where $Bits_I$, $Bits_P$ are bits of coded Intra (I) and Inter (P) frames, and QP_I, QP_P represent the average I-Frames and P-Frames quantization parameter. These values are also obtained directly from the encoding process and thus, do not increase the computational time or complexity of the method.

On the pixel level, noise and blur components (mean and ratio per feature) have been demonstrated to provide a good measure of degradations in a frame-by-frame assessment [41]. In the same way, blockiness [19, 42], described as a discontinuity between adjacent blocks in images and video frames [43], was demonstrated in our earlier study [14] to show promising results. Finally, measuring the inter-frame degradations becomes fundamental in the presence of network impaired video. To this end, temporal features such as the jerkiness (non-fluent and non-smooth presentation of frames) become fundamental [25]. Before they could be directly applied in the SL process, these eight NR metrics were averaged across the video and normalized between zero and one. Further details on how to compute these metrics are given in [14].

In addition to the video stream characteristics, we chose two network features (the received video bitrate and packet loss level) to capture the most significant transmission effects on video quality [34]. Intuitively, quality is related to bitrate (i.e. the number of bits received per time interval), whereby higher bitrates lead to better quality. However, this relation is highly non-linear, following a psychometric curve [44]. Earlier studies (some from us [45]) have shown how the parameters of the perception curve vary considerably across video types, compression values, bitrate etc. Bitrate is therefore a critical input to derive the prediction model. In addition, packet losses have been demonstrated to be the most impairing network conditions on video transmission systems [34, 8].

Thus, a measure of the packet loss level was included in the final feature set. These two parameters are calculated on the client side, during video reception, and are added to the other input features of the learning algorithm (i.e. the eight NR metrics).

These ten parameters conform the full characterization of the videos and serve both for training the SL model (offline on the server side) and for predicting the quality of the real-time received videos (in the client side, in the form of inputs to the trained SL method). Through SL, we derive the quality prediction model (i.e. the function $\hat{f}_{server}$ in Figure 2) by mapping input-output pairs of the training data. The model is then used to estimate the video quality, determining a suitable output value for any incoming stream (regardless of whether or not this has been part of the training set) [46].

Our method, as described in Figure 2, is generic and may be easily extended to explore different training features and benchmark quality (FR models or subjective studies), different video datasets and different SL algorithms. The details of our experimental evaluations, including the choice of the different SL algorithms are given next.

The flexibility, scalability and real-time characteristics of our method make it a very suitable candidate to close the feedback loop between client and server for service and network providers, who are nowadays encouraged to manage QoE through tailor-made adaptations of open systems/protocols (due to the changing network conditions and the exponential increase of users).

4. Evaluation Methodology

We describe here the complete methodology used to evaluate the predictive quality metric introduced in Section 3. The experimental test-bed (Section 4.1, Figure 3) comprises all the components used to carry out a comparative evaluation with the benchmark quality metric. The prediction model (i.e. the $\hat{f}_{server}$) is computed offline (as per Figure 2a), exploring a whole range of machine learning options, as detailed in Section 4.2. Our method is generic, it does not demand a specific learning or benchmark algorithm. We have adopted VQM as our benchmark, which is broadly used when subjective assessment is not viable, as it is the case of commercial live streaming services.

4.1. Experimental Test-bed

Once the quality prediction function ($\hat{f}_{server}$) has been computed (offline), we are ready to perform real-time streaming tests, based on the components depicted in Figure 3. Our testbed allows streaming any of the dataset videos between the server and the client. We used an RTP video server to handle the streaming process, and a commercial network emulator (PacketStorm Hurricane II)[2] to shape and impair the stream in a controlled (replicable) environment.

[2]http://packetstorm.com/packetstorm-products/hurricane-ii-software/

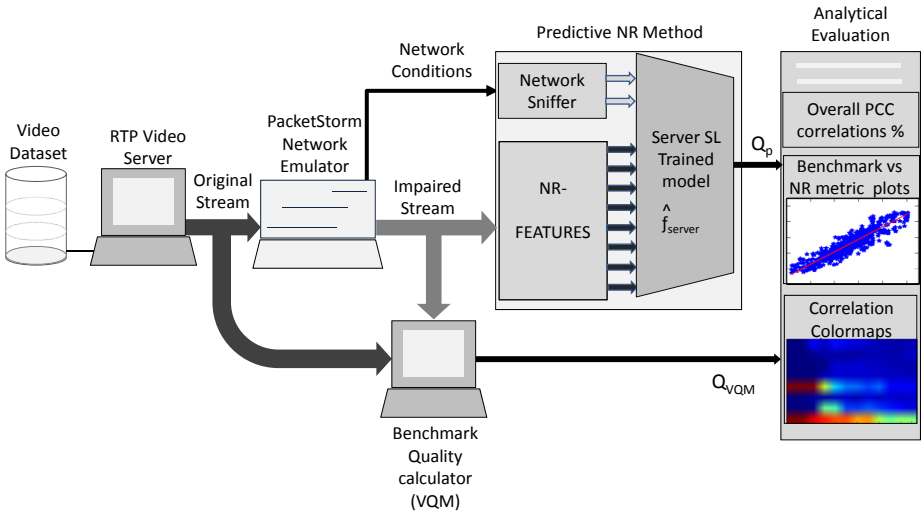


Figure 3: Evaluation test-bed.

The network-impaired stream is then fed to our client application, which generates the predicted metric Q_p. In parallel, we generate the benchmark quality index Q_{vqm}. We stream all videos, in turn, under a range of network conditions (Table 1), obtaining a full range of quality values, ready for statistical analysis. The accuracy is measured by means of a Pearson correlation (PCC) [35] between the predicted quality and the benchmark quality.

As benchmark quality we selected VQM because of its demonstrated good correlation to the human vision system and to subjective feedback [10, 11]. Furthermore, in [14] we characterized the whole dataset by means of its VQM index, showing its suitability as a benchmark.

4.2. Supervised learning methodology

Given the broad variety of ML approaches in the literature, an important element of our work was to explore different algorithms and find suitable avenues. To this end, our experimental framework (Figure 3) is sufficiently generic to perform tests on any type of SL algorithms (we have not included unsupervised learning methods in our study).

Among the well-established SL methods, we started experimenting with 16 different ones, ending up with a selection of 9 methods that cover a representative set of algorithms, ranging from the least complex (towards the top of Table 3) to the most complex ones (towards the bottom of Table 3). Methods may be broadly categorized in two. On the one hand, the white-box methods are able to capture a comprehensible relation between input and output features and thus, can be interpreted in a straightforward way by a human operator. On the other hand, black-box methods do not offer such relation and do not help understanding how certain predictions are derived. We review below the key features of the methods under scrutiny.

One of the most known and simplest white boxes is linear regression [47], which attempts to model the relationship between a scalar (output) and one or more independent variables by means of a linear multidimensional model of the input data.

Decision trees learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value [48]. They are classified according to the type of output provided.

On the one hand, tree models, where the target variable takes a value from a finite set, are called classification trees. Leaves represent class labels and branches, conjunctions of features that lead to those class labels. On the other hand, decision trees, where the target variable can take continuous values (typically real numbers), are called regression trees.

The performance of regression and decision trees can be further improved by means of an ensemble approach. Ensembles use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms [49]. Evaluating the prediction of an ensemble typically requires more computation than evaluating the prediction of a single model. Thus ensembles are mostly used as a way to compensate for poor learning algorithms by performing extra computation. For this reason, fast (less accurate) algorithms such as decision trees are commonly used with ensembles.

Since the first conception, several approaches to combine the ML models have appeared. One early method is the Bootstrap aggregating [50], often abbreviated as Bagging, which involves having each model in the ensemble vote with equal weight. Another method, Boosting [51], involves incrementally building an ensemble by training each new model instance to emphasize the training instances that previous models misclassified. In some cases, boosting has been shown to yield better accuracy than bagging, but it also tends to be more likely to over-fit the training data.

The most common implementation of Boosting is Adaboost [52]. In Adaboost, short for "Adaptive Boosting", the output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. While specific learning algorithms will tend to suit some particular problem types better than others, and will typically have many different parameters and configurations to be adjusted before achieving optimal performance on a dataset, Adaboost (with decision trees as the weak learners) is often referred to as the best out-of-the-box classifier. Adaboost is used only for classification, thus in order to use it, the quality index range (0 to 1) needs to be converted into a finite set of values.

Another type of Boosting known to work very well together with regression trees is LS-Boost (least squares) [53]. Like other Boosting methods, LS-Boosting combines weak learners into a single strong learner, in an iterative fashion, where the goal is to learn the model that predicts the outputs while minimizing the mean squared error to the true values (averaged over the training set).

White boxes are appreciated for their comprehensive models. Yet, they have also been demonstrated to have limited predictive capacity or to be inflexible and computationally cumbersome. The best classification and regression accuracy

is typically achieved by black-box models such as Gaussian processes or neural networks, or complicated ensembles of them [54]. These models do not, in general, provide a clear explanation of the reasons as to how they have come to a certain prediction.

The Gaussian Process Regression (or Kriging) [55] provides an example. The basic idea of Kriging is to predict values by means of interpolation in which the interpolated values are modeled by a Gaussian process governed by prior covariances. Under suitable assumptions on the priors, Kriging gives the best linear unbiased prediction of the intermediate values.

Support Vector Machines (SVMs) [56]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training samples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new samples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the samples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New samples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall in.

Finally, we tested artificial neural networks (ANNs) [57], a family of models inspired to biological neural networks, used to estimate or approximate functions that can depend on a large number of generally unknown inputs. ANNs are generally presented as systems of interconnected "neurons" which exchange messages between each other. The connections have numeric weights that can be tuned based on various optimization methods, making neural nets adaptive to inputs and capable of learning. The feedforward neural network was the first and simplest type of artificial neural network devised. In this case, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. A variation on the feedforward network is the cascade forward network which has additional connections from the input to every layer, and from each layer to all following layers.

We implemented these methods based on the ML toolbox [58] and the Neural Network toolbox [59] of Matlab, and the library LIBSVM [60] for the support vector regression model. Each algorithm requires the tuning of certain parameters in order to optimize their performance. The values included in Table 3 (fourth column), have been found to perform better with our dataset. In order to perform the Multiple Linear Regression, we added a bias vector (a vector of all ones) to the input data. As we explained in the previous section, to use the ensemble decision tree with Adaboost, the dataset outputs have to be converted to a set of finite values. After careful experimentation, we set the number of classification classes to 100, ranging for 0.00 to 0.99. Values are then rounded to their second decimal.

Another important choice in performing ML experiments consists of the way the training set is picked out of the whole dataset. The method used is bound to have a sensitive effect on the performance of the prediction models and, ultimately, on the accuracy of the NR metric. To mimic typical situations faced

Table 3: Parameters used for the different machine learning techniques

Type	Technique	Acronym	Parameters
W	Multiple Linear Regression	LR	Added bias
H I T	Standard Regression Tree	RT	type:binary N. Branches>15
E	Ensemble Regression Tree LS-Boost	ERT-LSB	N. Models:500 N. Branches>15 Learning Rate:0.01
B O X	Ensemble Regression Tree Bagging	ERT-BR	N. Models:500 N. Branches>15
	Ensemble Decision Tree Adaboost	EDT-AB	N. Classes: 100 (1/100) N. Models:200 N. Branches>10 Learning Rate: 0.2
B L	Gaussian Process Regression	GPR	Method: exact Basis: constant Kernel: squaredexponential
A C K	Support Vector Regression	SVR	type: epsilon kernel: radial basis cost and epsilon: 20 & 0.1
B O X	FeedForward Neural Network	FNN	N. Hidden Neurons: 20 Training: Levenberg -Marquardt
X	Cascaded FW. Neural Network	CNN	N. Hidden Neurons: 20 Training: Levenberg -Marquardt

by a video service provider, we carried out two set of experiments. Blind prediction, represents the worst-case performing scenario, whereby the video under consideration is unknown to the machine learning model (Section 5). In the most common scenario, the video server is able to prepare the ML model from samples of the whole data-set before being transmitted to the client (Section 6). In this way, characterizations of all the videos are present in the model of the system. For it, we consider the performance of these type of cases using random cross-validation tests (Section 6). Finally, we studied the sensitivity of our metric to the size of the training set (Section 7) and the computational time required for our approach compared to the FR benchmarks (Section 8).

5. Evaluation of the worst-case scenario: unknown video class, blind prediction

As mentioned in Section 3, on service launch the service provider will have a representative video types set and thus an initial model can be constructed and sent to the client. When, due to a completely new type of video, the prediction model is to be updated, the server will notify the client and the model in the client will be upgraded.

Therefore, the most typical scenario will see an up-to-date prediction model.

Table 4: Overall performance of nine machine learning algorithms in blind mode (worst-case scenario, 10-fold cross-validation). Values indicate PCC correlations to VQM, averaged for each video type across all compression levels and network conditions (96 cases). Cell colors give qualitative correlation levels: green (best); orange (median); and red (worst).

V.T.	LR	RT	ERT-LSB	ERT-BR	EDT-AB	GPR	SVR	FNN	CNN
bs1	0.813	0.86	0.95	0.95	0.74	0.832	0.564	0.94	0.956
mc1	0.89	0.843	0.9277	0.8952	0.5842	0.8668	0.4928	0.8851	0.9198
pa1	0.87	0.955	0.9706	0.916	0.8199	0.9342	0.7085	0.9213	0.9542
pr1	0.89	0.69	0.7684	0.8185	0.2858	0.8684	0.2439	-0.6188	-0.2887
rb1	0.89	0.7067	0.901	0.9495	0.5063	0.8086	0.7054	0.4479	0.868
rh1	-0.32	0.783	0.7972	0.772	-0.3569	-0.1172	0.725	-0.0294	-0.2758
sf1	0.94	0.929	0.9729	0.9743	0.7615	0.9542	0.7462	0.9152	0.9531
sh1	0.85	0.828	0.9206	0.9294	0.6161	0.9267	0.7582	0.6712	0.8667
st1	0.94	0.858	0.9705	0.9665	0.7082	0.9634	0.4661	0.9673	0.8204
tr1	0.92	0.859	0.96	0.96	0.5084	0.976	0.711	0.9376	0.9441
All	0.768	0.83	0.9147	0.9137	0.517	0.8013	0.6121	0.6039	0.6718
	±0.383	±0.085	±0.074	±0.0678	±0.344	±0.328	±0.168	±0.533	±0.505

This case will be evaluated in Section 6. We now consider the worst-case scenario, to evaluate the bottom-line performance of our metric. To test SL in blind mode, the model is trained with nine (out of ten) video types and is tested on the 96 samples of the remaining one (8 compression levels and 12 network conditions). For statistical significance, we performed a 10-fold cross-validation test, evaluating in turn, each of the ten videos as a new (unknown) class.

The overall performance of the 9 different machine learning algorithms in blind mode is detailed in Table 4. The first striking result is that our metric always performs considerably better than any of the conventional NR metrics (Table 2). The reason for this comes from the fact while each of the individual NR features analyses the video in one single aspect, an SL approach combines the action of the whole range of metrics and network conditions to provide an assessment. The worst-case performance of the worst-performing machine learning algorithms (51.7% EDT-AB Table 4) was better than the best-performing NR metric (41.7% CX Table 2). The Ensemble Regression Trees methods achieve the best average performance of 91.3% (ERT-BR) and 91.4% (ERT-LSB).

Comparing the different machine learning algorithms, we found another important result: the white-box approaches (LR, RT, ERT-BR, ERT-LSB and EDT-AB) outperform the black-box ones (GPR, SVR, FNN and CNN). This is interesting because the former methods tend to be less computationally intensive. Intuitively, we can explain this result by looking at the standard deviations, which tend to be rather large (up to 53% in FNN). This is to be expected in blind prediction when the samples are significantly different. In fact, the most distinctive videos (the ones with distinctive time and space complexity) were predicted with lower accuracy. For instance, video type pa1 is well represented by the other nine video types: thus the 10-fold validation for pa1 leads to consistently accurate predictions (71 to 93%). At the other end of the spectrum is video type pr1, which leads to diverse prediction accuracies (-62% to 87%). We must stress that these high variations are typical of blind prediction and will not appear in the most common operational condition (Section 6).

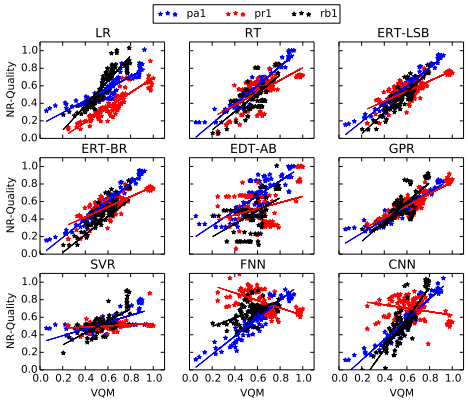
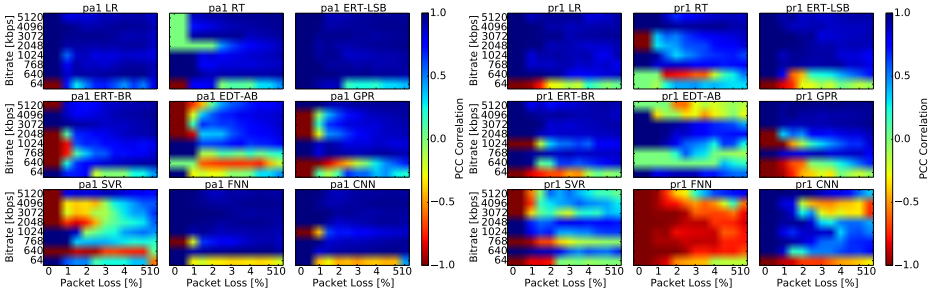
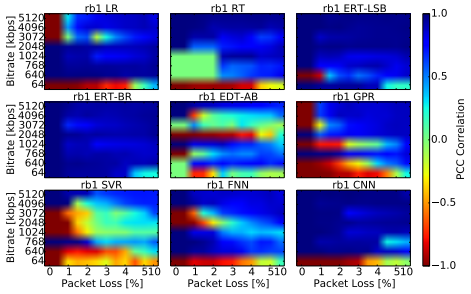


Figure 4: Correlation diagrams of nine different prediction algorithms (LR, RT, ERT-LSB, ERT-BR, EDT-AB, GPR, SVR, FNN, CNN) in comparison to VQM (used as benchmark). The three sample videos are: pa1 (blue stars); rb1 (black stars); and pr1 (red stars).



(a) Correlation maps for video pa1.

(b) Correlation maps for video pr1.



(c) Correlation maps for video rb1.

Figure 5: Pearson correlation to VQM of the nine prediction algorithms (LR, RT, ERT-LSB,ERT-BR, EDT-AB, GPR, SVR, FNN, CNN), considering bitrates between 64 and 5,120 Kbps and packet losses between 0 and 10%. Video types: a) Pedestrian Area (pa1); b) Park run (pr1); and c) River bed (rb1).

To better explore the differences across the test videos, Figure 4 shows the correlation diagrams of the three most distinctive videos (pa1, pr1 and rb1), whose NR metrics were scrutinized in Section 2 (Figure 1). Each diagram picks

one machine learning algorithm in relation to the benchmark VQM, showing the three video types in different colors. In this way, the most accurate predictions are concentrated around the main diagonal (y=x). We observe how video type pa1 (blue stars) is predicted consistently well, followed by rb1 (black stars). On the other hand, pr1 (red stars) is the most difficult to predict. Overall, RT and ERT-LSB are the ones that deal the best with blind prediction; and in general, black box approaches perform the worst. Of these, only GPR performs consistently well on all videos, while the two neural networks (FNN and CNN) struggle with rb1 and fail with pr1. The support vector machine fails on all cases.

Finally, to visualize the working range of the different machine learning algorithms, Figure 5 shows the Pearson Correlation (PCC) colormaps analogous to those of Figure 1 (NR metrics). Strikingly, the well-correlated range (dark blue) extends much further (both in packet loss and bitrate levels) than the original NR metrics. The color patterns show also how the less complex machine learning methods (the upper maps in Figures 5a, 5b and 5c) have a broader operational range than the more complex algorithms (lower maps in Figure 5a, 5b and 5c). As we already hinted, the best performers are the Ensemble Regression Trees, particularly LS-Boost (ERT-LSB) achieves nearly full correlation for all bitrates and network conditions.

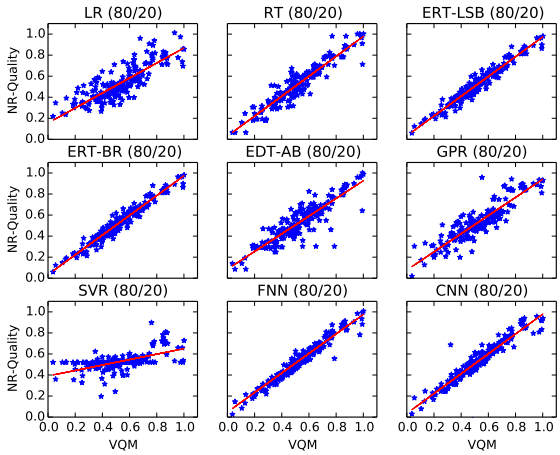
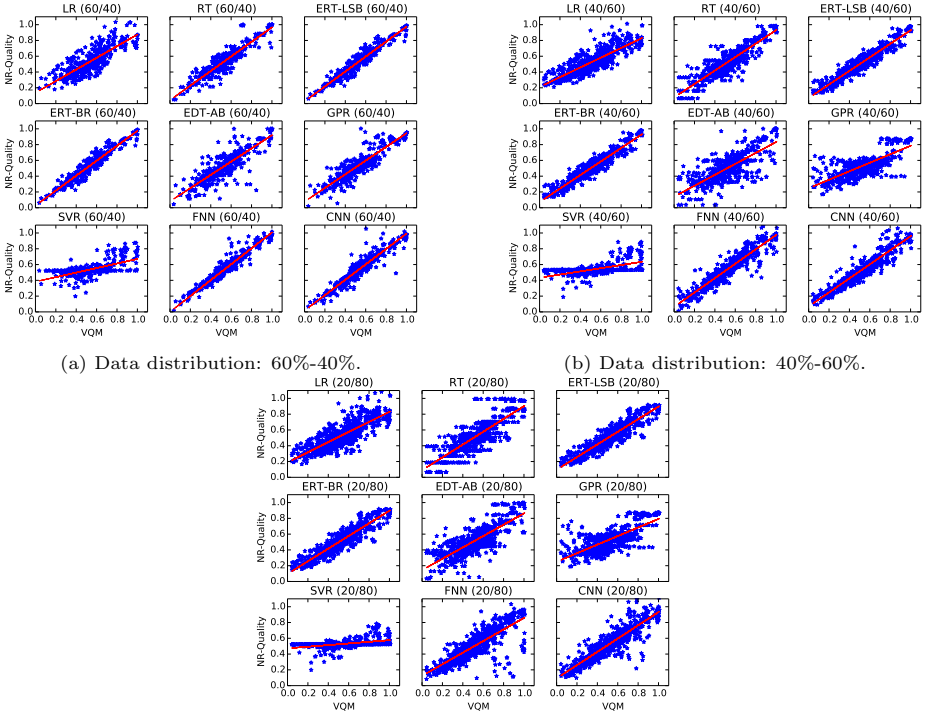


Figure 6: 80%-20% training to testing data distribution. The diagrams show the overall correlation diagrams of nine different prediction algorithms (LR, RT, ERT-LSB, ERT-BR, EDT-AB, GPR, SVR, FNN, CNN) in comparison to VQM (used as benchmark).

6. Evaluation of common-case scenario: known video class, prediction based on prior video traces

We evaluate here the typical scenario in which our prediction based metric is assessed on video conditions (type, rate and packet loss level) that have previously been seen by the SL algorithm. Thus, we can assume that the prediction



(a) Data distribution: 60%-40%.

(b) Data distribution: 40%-60%.

(c) Data distribution: 20%-80%.

Figure 7: Predicted values vs benchmark quality (VQM) for different distributions of training and testing data: a) 60%-40%; b) 40%-60%; c) 20%-80%.

model will have been trained on samples from all the videos belonging to the service provider's dataset. Our aim is to evaluate the performance of our metric (as described in Section 3) under realistic conditions, for a representative set of machine learning algorithms (LR, RT, ERT-LSB, ERT-BR, EDT-AB, GPR, SVR, FNN, CNN).

We follow a standard machine learning evaluation method. We randomize the whole dataset (960 samples), splitting it into five subsets (192 samples each). On each of the nine machine learning algorithms, we perform a 5-fold cross-validation test, using in turn one subset for testing and the other four for training. Just like in blind prediction (Section 5), the resulting nine prediction models are used to find Pearson Correlations with VQM, along with averages and deviation values.

The first set of results is included in Table 5 (first row) and depicted in Figure 6. We notice a definite improvement compared to blind prediction (Table 4 and Figure 4). If we exclude SVR, that has the smallest correlation to VQM ($63\% \pm 3$), all other prediction algorithms are consistently accurate, in terms of both correlations to VQM (in the $78 - 97\%$ range) and deviations (in the $0.4 - 6\%$ range). Even more remarkably, all our prediction-based metrics

Table 5: Overall performance of nine machine learning algorithms, for different sizes of the training and testing datasets. Values indicate overall PCC correlations to VQM (and standard deviations). Cell colors give qualitative correlation levels: green (best); orange (median); and red (worst).

TR/TE	LR	RT	ERT-LSB	ERT-BR	EDT-AB	GPR	SVR	FNN	CNN
80/20	0.78 ±0.06	0.93 ±0.01	0.97 ±0.004	0.95 ±0.006	0.83 ±0.04	0.888 ±0.02	0.634 ±0.04	0.92 ±0.04	0.89 ±0.15
60/40	0.77 ±0.04	0.925 ±0.004	0.97 ±0.003	0.94 ±0.006	0.783 ±0.07	0.824 ±0.041	0.62 ±0.01	0.93 ±0.05	0.825 ±0.19
40/60	0.77 ±0.03	0.9 ±0.01	0.9561 ±0.002	0.92 ±0.011	0.8 ±0.0377	0.78 ±0.02	0.57 ±0.012	0.92 ±0.016	0.76 ±0.12
20/80	0.76 ±0.03	0.85 ±0.02	0.93 ±0.01	0.87 ±0.01	0.74 ±0.07	0.75 ±0.02	0.47 ±0.03	0.86 ±0.04	0.85 ±0.02

work on the whole range of network conditions (0-10% packet loss) and bitrates (64kbps to 5.12Mbps). We can confidently claim so thanks to the low deviations reached when averaging across all network conditions (0.4-6% range).

7. Performance vs size of the training dataset

Having established the accuracy of prediction-based metrics across a variety of machine learning methods, our next aim was to explore how the size of the training dataset affected the metrics accuracy. In other words, how many video conditions would a service provider have to use to train accurate predictions models?

To this end, we followed the same evaluation method of Section 6, splitting the 960-sample dataset in five subsets and performing a 5-fold cross-validation test. However, this time we evaluated the machine learning algorithms on different training sample sizes. Figure 7 and Table 5 capture all the results, considering training and testing samples of (80%;20%), (60%;40%), (40%;60%) and (20%;80%), respectively. As expectable, the reduction of the training set leads to an increase in error. However, this is comparably small. Overall, when the training set is reduced from 80% to 60%, 40% and 20%, the accuracy drops by an average of 2.4%, 4.7% and 7.9%, respectively. For instance, if we look at our 960-sample dataset we can expect an overall accuracy in the area of 86.6% (using 768 samples for training), 84.1% (using 576 samples) and 78.7% (using 192 samples).

Assessing several machine learning approaches is very useful in pinpointing the most effective algorithms and, in turn, pursue even better performance. For instance, neural networks show a consistent performance in excess of 85%, even when the training set is reduced down to 20%. The best performers are the Ensemble Regression Trees, particularly LSB with its 97% accuracy (with 80% training samples) that drops only to 93% (with 20% training samples). ERT-LSB is also the best performer on blind predictions (91% overall accuracy, Table 4), which makes this the algorithm of preference for our predictive NR method.

Table 6: Overall computational time (in seconds) for the training of nine machine learning algorithms, for different sizes of the training and testing datasets. Cell colors give completion performance: green (best); orange (median); and red (worst).

TR/TE	LR	RT	ERT-LSB	ERT-BR	EDT-AB	GPR	SVR	FNN	CNN
80/20	0.0003	0.012	4.48	3.63	98.5	3.7	0.03	0.45	0.44
	±1e−4	±1e−4	±0.02	±0.01	±16.8	±2.32	±0.002	±0.04	±0.03
60/40	0.0002	0.01	4.04	4.1	51.2	1.8	0.02	0.38	0.38
	±3e−4	±6e−4	±0.01	±0.1	±9.5	±1.05	±8e−4	±0.01	±0.03
40/60	0.0001	0.008	3.64	3.23	20.6	0.3	0.007	0.37	0.36
	±2e−5	±1e−4	±0.01	±0.002	±2.5	±0.04	±5e−4	±0.01	±0.005
20/80	0.0001	0.007	3.24	3.04	5.4	0.07	0.002	0.3	0.32
	±1e−5	±1e−4	±0.004	±0.01	±0.16	±0.01	±2e−4	±0.01	±0.02

8. Computational Trade-offs

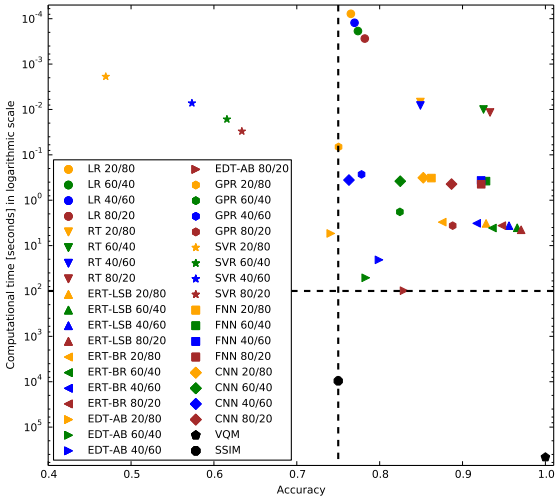


Figure 8: Performance trade-offs (accuracy and computational time) of the different learning algorithms (colored symbols), considering all training/testing combinations. SSIM and VQM (in black) are used to benchmark our metrics. With the exception of SVR, all other learning models perform much better than SSIM. Computational time (in log scale) is 4 orders smaller than SSIM and 6 orders smaller than VQM.

The models of our performance metric are trained in the background (Figure 2, top), before being used in the client (Figure 2, bottom). Thus, the running time of the learning algorithms will not affect the real-time quality metric computational times. Still, it is interesting to see the trade-offs achievable with the different machine learning techniques, as these will affect the service provider's ability to manage video datasets at scale.

To this end, we follow the same evaluation method of Section 7, splitting the 960-sample dataset in five subsets, performing a 5-fold cross-validation test and evaluating the algorithms on the four training-testing subdivisions considered earlier, i.e. 80%-20%; 60%-40%; 40%-60%; 20%-80%. In each case, we measure

the time incurred to train the model. We perform this process on a Laptop (HP EliteBook) with an Intel Core i7 processor and 7,7GB of RAM memory.

As it could be expected, smaller training sets incur faster completion times (Table 6). However, the difference is not significant (computation time orders do not vary between the 20/80 split and the 80/20 split). The fastest algorithm was LR, with computation time in the millisecond scale. Interestingly, this is not the least accurate metric (overall 77% correlation, Table 5).

On the other end of the range, the ensemble regression trees (ERT-BR and ERT-LSB) incur times ranging from 3 to 4.5 seconds. This is because they have to build 500 consecutive models before they can complete the trained models. Yet, these lead to the most significant accuracy.

Even when deployed in on a low-spec laptop, the computational times of the prediction metrics are negligible and, certainly, compatible with the typical background processes of a service provider. Also, in a commercial setting the background QoE processes will be supported by dedicated servers and, when necessary, data centers or cloud services. Hence, the times involved in characterizing the video dataset would not constitute a bottleneck.

Figure 8 shows the performance trade-offs (accuracy and computational time) of the different learning algorithms (colored symbols), considering all training/testing combinations. To benchmark our metrics, we include also SSIM and VQM (in black). With the exception of SVR, all other learning models perform much better than SSIM, although the former are NR and the latter is FR. Of crucial importance is our finding of the learning computational times, which are four orders smaller than SSIM and six orders smaller than VQM. Thus prediction metrics are comparably as accurate as VQM while scaling significantly better.

9. Related work on Machine Learning for NR Quality Assessment

In our previous research, we have conducted a range of preliminary studies that have provided basis and motivations to the present paper. Our most relevant works are summarized next. Our earlier attempts to develop NR metrics based on conventional features (i.e. without using machine learning), lead to a formula that combined scene complexity and motion and could be computed in real-time [38]. At the same time, we were exploring the use of machine learning to address fundamental limitations of conventional NR metrics [14], mainly the lack of generality and poor performance. In [61] we showed the use of Reinforcement Learning to optimize video quality in adaptive streaming, without using complex heuristics. In [1] we showed how artificial neural networks could determine a linear combination of blur and noise that performed significantly better than these two NR metrics in isolation. Finally, our recent survey of machine learning in NR video quality assessment [62] provides a snapshot of the state-of-the-art on which our work is based. A selection of the most relevant on-going efforts is briefly described below.

In the last decade, several researchers have explored the machine learning path in order to improve both the generality and accuracy of NR metrics. Al-

ready in 2002, Gastaldo et al. introduced one of the first methods to estimate the video quality using artificial neural networks [63]. They proposed the use of circular back propagation networks (based on bitstream layer parameters) in order to mimic the users perception of compressed MPEG2 videos. Their approach showed promising results on a 12-video dataset from the motion picture expert group (MPEG). Their study focused on video distortions deriving merely from compression and explored a specific machine learning method.

Also working on compressed videos, Le Callet et al. [64] employed an interesting convolutional neural network as a Reduced Reference (RR) method to allow a continuous-time quality estimation and scoring of the video. Unlike our NR approach, in which the server transmits the machine learning model updates only on service launch and in the case that an update is due, their method (as any RR metric) requires the transmission of features extracted from the original video together with the video under scrutiny.

Zhu et al. [65] proposed the use of neural networks and features extracted from the analysis of Discrete Cosine Transform (DCT) coefficients of each decoded frame from a video sequence to predict its quality. Their approach showed good correlation results in compressed videos of four different well-known datasets. However, their method is distortion specific, and thus of a more limited scope than our case. Furthermore, the complexity of the approach makes it not viable to real-time deployments.

Staelens et al. [66] presented an NR video quality estimation method which uses a symbolic regression framework trained on a large set of parameters extracted from the codec. While obtaining good correlation with subjective tests, their approach is suited only to H.264 compressed streams, thus loosing on generality.

Similar principles were proposed in [67] by using features extracted from specific codecs (MPEG or H.264/AVC), the analysis of DCT coefficients, the estimation of the quantization level used in the I-frames to measure quality of videos distorted by only the compression process. They show high correlation with some state-of-the-art metrics (FR, RR and NR). However, their approach is only suited to a specific type of codec and the complexity of the feature extraction process makes this NR metric incompatible with real-time applications.

Shahid et al. [28] proposed a model combining different bitstream-layer features using an Artificial Neural Network to estimate the quality. They tested their method on compressed videos but focused on correlations with PSNR.

The key differentiator between our work and other valuable on-going efforts is our focus on a generic learning framework for assessing end-to-end streaming in real-time. Our predictive method (Figure 2) and evaluation methodology (Figure 3) are generic - i.e. completely independent from type of video, compression, benchmarking quality, transmission means and machine learning algorithm. We place the heavy part of the machine learning (training) on a background process, allowing for a light-weight evaluation metric to be executed in real-time, even on thin clients. We do not have to rely on synthetic impairments and have a system that can be employed in a typical video service provisioning platform or for real-time quality management.

10. Conclusion

Low-Computational No-Reference video quality methods have the potential to provide real-time video quality assessment and automated quality control, as required by services such as video streaming, video on demand, and network management. In these situations, both subjective assessment and computationally-intensive objective methods are unfeasible. At the same time, simplistic NR methods would be functionally and computationally viable but fail to deliver accurate results, as we demonstrated in our previous work [14], and, specifically, cannot handle network-impaired streams. On the other hand, existing NR methods based on machine learning tend to be heavyweight and often lack generality.

In this work, we introduce a generic machine learning framework (Figure 2) that allows deriving a predictive NR assessment metric. We explore the efficiency and accuracy of our metric for a broad representation of supervised-learning techniques (Table 3), using a varied video dataset (Table 1).

Through an extensive analysis (Section 5 to 8), we demonstrated how our approach is not tied to any particular type of video, compression, or transmission means. In fact, the metric performance remains remarkably high even when the training set is reduced from 80 to 20% (Table 5), indicating that models can accurately predict 80% of unknown conditions.

We are particularly keen to have developed an NR metric that operates accurately under lossy networks. We tested the whole 0-10% packet loss range, which reflects the most extreme Internet conditions. Overall, we have achieved an over 97% correlation to VQM, demonstrating that it is possible to develop an NR metric that is as accurate as an FR method, while allowing real-time assessment of video quality in realistic streaming scenarios.

This NR method is meant as a light-weight means to close the feedback loop between client and server. We envision our NR method to be applicable to client-driven adaptive streaming and video-on-demand. Furthermore, we aim to apply it to the prominent scenario that network and service providers face today, whereby they can measure Quality of Service but don't have feedback about the Quality of Experience that is actually delivered to the end-user (end device).

Acknowledgment

This work has been carried out in the context of the European Research Council project BROWSE (Beam-steered Reconfigurable Optical-Wireless System for Energy-efficient communication - Grant 291632) and the ICT COST Action 3D-ConTourNet (IC1105).

References

[1] M. Torres Vega, E. Giordano, D. C. Mocanu, A. Liotta, Cognitive no-reference video quality assessment for mobile streaming services, in: in proc.

of the 7th International Workshop on Quality of Multimedia Experience (QoMex), 2015. doi:10.1109/QoMEX.2015.7148128.

[2] I. Politis, L. Dounis, T. Dagiuklas, H.264/svc vs. h.264/avc video quality comparison under qoe-driven seamless handoff, Signal Processing: Image Communication 27 (8) (2012) 814–826. doi:10.1016/j.image.2012.01.006.

[3] E. Ekmekcioglu, G. C. Gurler, A. Kondoz, A. M. Tekalp, Adaptive multi-view video delivery using hybrid networking, IEEE Trans. Circuits and Systems for Video Technology 2016.

[4] L. Atzori, A. Floris, G. Ginesu, D. Giusto, Streaming video over wireless channels: Exploiting reduced-reference quality estimation at the user-side, Image Commun. 27 (10) (2012) 1049–1065. doi:10.1016/j.image.2012.09.005.

[5] A. Ahmad, A. Floris, L. Atzori, Qoe-aware service delivery: A joint-venture approach for content and network providers, in: Eighth International Conference on Quality of Multimedia Experience, QoMEX 2016, Lisbon, Portugal, June 6-8, 2016, 2016, pp. 1–6. doi:10.1109/QoMEX.2016.7498972.

[6] K. Panetta, L. Bao, S. S. Agaian, A human visual "no-reference" image quality measure., IEEE Instrum. Meas. Mag. 19 (3) (2016) 34–38.
URL http://dblp.uni-trier.de/db/journals/imm/imm19.html#PanettaBA16

[7] P. Paudyal, Y. Liu, F. Battisti, M. Carli, Video quality of experience metric for streaming services, in: Image Processing: Algorithms and Systems XIV, San Francisco, California, USA, February 14-18, 2016, 2016, pp. 1–5.
URL http://ist.publisher.ingentaconnect.com/contentone/ist/ei/2016/00002016/00000015/art00004

[8] P. Paudyal, F. Battisti, M. Carli, Impact of video content and transmission impairments on quality of experience, Multimedia Tools and Applications 2016. doi:10.1007/s11042-015-3214-0.

[9] M. H. Pinson, S. Wolf, A new standardized method for objectively measuring video quality, IEEE Transactions on broadcasting 50 (3) (2004) 312–322. doi:10.1109/TBC.2004.834028.

[10] S. Chikkerur, V. Sundaram, M. Reisslein, L. J. Karam, Objective video quality assessment methods: A classification, review, and performance comparison., IEEE Transactions on Broadcasting 57 (2) (2011) 165–182. doi:10.1109/TBC.2011.2104671.
URL http://dblp.uni-trier.de/db/journals/tbc/tbc57.html#ChikkerurSRK11

[11] M. Vranješ, S. Rimac-Drlje, K. Grgić, Review of objective video quality metrics and performance comparison using different databases, Image Commun. 28 (1) (2013) 1–19. doi:10.1016/j.image.2012.10.003.

[12] K. Pandremmenou, M. Shahid, L. P. Kondi, B. Lövström, A no-reference bitstream-based perceptual model for video quality estimation of videos affected by coding artifacts and packet losses, in: Human Vision and Electronic Imaging XX, San Francisco, California, USA, February 9-12, 2015, 2015, p. 93941F. doi:10.1117/12.2077709.

[13] Z. Chen, N. Liao, X. Gu, F. Wu, G. Shi, Hybrid distortion ranking tuned bitstream-layer video quality assessment, IEEE Trans. Circuits Syst. Video Techn. 26 (6) (2016) 1029–1043. doi:10.1109/TCSVT.2015.2441432.

[14] M. Torres, V. Sguazzo, D. C. Mocanu, A. Liotta, An experimental survey of no-reference video quality assessment methods, Int. J. Pervasive Computing and Communications 12 (1) (2016) 66–86. doi:10.1108/IJPCC-01-2016-0008.

[15] M. Shahid, A. Rossholm, B. Lövström, H. Zepernick, No-reference image and video quality assessment: a classification and review of recent approaches, EURASIP J. Image and Video Processing 2014 (2014) 40. doi:10.1186/1687-5281-2014-40.

[16] Q. Huynh-Thu, M.-N. Garcia, F. Speranza, P. Corriveau, A. Raake, Study of rating scales for subjective quality assessment of high-definition video., IEEE Transactions on Broadcasting 57 (1) (2011) 1–14.
URL http://dblp.uni-trier.de/db/journals/tbc/tbc57.html#Huynh-ThuGSCR11

[17] U. Engelke, H.-J. Zepernick, Perceptual-based quality metrics for image and video services: A survey, in: Next Generation Internet Networks, 3rd EuroNGI Conference on, 2007. doi:10.1109/NGI.2007.371215.

[18] B. Staehle, A. Binzenhöfer, D. Schlosser, , B. Boder, Quantifying the influence of network conditions on the service quality experienced by a thin client user, in: 14. GI/ITG Konferenz Messung, Modellierung und Bewertung von Rechen- und Kommunikationssystemen (MMB 2008), Dortmund, Germany, 2008.

[19] C. Perra, A low Computational Complexity Blockiness Estimation Based on Spatial Analysis, in: in IEEE 22nd Telecommunications Forum, 2014. doi:10.1109/TELFOR.2014.7034606.

[20] S. Winkler, P. Mohandas, The evolution of video quality measurement: From PSNR to hybrid metrics, IEEE Transactions on Broadcasting 54 (3) (2008) 660–668. doi:10.1109/tbc.2008.2000733.

[21] E. Lehmann, G. Casella, Theory of Point Estimation, Springer Verlag, 1998. doi:10.1007/b98854.

[22] Z. Wang, L. Lu, A. C. Bovik, Video quality assessment based on structural distortion measurement, Signal Processing: Image Communication 19 (2) (2004) 121–132. doi:10.1016/s0923-5965(03)00076-6.

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image Quality Assessment: From Error Visibility to Structural Similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612. doi:10.1109/tip.2003.819861.

[24] A. G. Ciancio, A. L. N. T. da Costa, E. A. B. da Silva, A. Said, R. Samadani, P. Obrador, No-reference blur assessment of digital pictures based on multifeature classifiers., IEEE Transactions on Image Processing 20 (1) (2011) 64–75. doi:10.1109/TIP.2010.2053549.

[25] S. Borer, A model of jerkiness for temporal impairments in video transmission, in: in proc. of the Second International Workshop on Quality of Media Experience (QoMEX), 2010. doi:10.1109/QOMEX.2010.5516155.

[26] T. Brandão, M. Queluz, No-reference quality assessment of h.264/avc encoded video, IEEE Trans. on Circuits and Systems for Video Tech. 20 (11) (2010) 1437–1447. doi:10.1109/TCSVT.2010.2077474.

[27] J. P. Vink, G. de Haan, No-reference metric design with machine learning for local video compression artifact level, J. Sel. Topics Signal Processing 5 (2) (2011) 297–308. doi:10.1109/JSTSP.2010.2055832.

[28] M. Shahid, J. Panasiuk, G. V. Wallendael, M. Barkowsky, B. Lövström, Predicting full-reference video quality measures using HEVC bitstream-based no-reference features, in: Seventh International Workshop on Quality of Multimedia Experience, QoMEX 2015, Pilos, Messinia, Greece, May 26-29, 2015, 2015, pp. 1–2. doi:10.1109/QoMEX.2015.7148118.

[29] Y. Xue, B. Erkin, Y. Wang, A novel no-reference video quality metric for evaluating temporal jerkiness due to frame freezing, CoRR abs/1411.1705.
URL http://arxiv.org/abs/1411.1705

[30] T. Shanableh, A regression-based framework for estimating the objective quality of HEVC coding units and video frames, Signal Processing: Image Communication 34 (2015) 22–31. doi:10.1016/j.image.2015.02.008.

[31] Y. J. Liang, J. G. Apostolopoulos, B. Girod, Analysis of Packet Loss for Compressed Video: Effect of Burst Losses and Correlation Between Error Frames, Circuits and Systems for Video Technology, IEEE Transactions on 18 (7) (2008) 861–874. doi:10.1109/tcsvt.2008.923139.
URL http://dx.doi.org/10.1109/tcsvt.2008.923139

[32] F. Yang, S. Wan, Q. Xie, H. R. Wu, No-reference quality assessment for networked video via primary analysis of bit stream, IEEE Trans. Circuits Syst. Video Techn. 20 (11) (2010) 1544–1554. doi:10.1109/TCSVT.2010.2087433.
URL http://dx.doi.org/10.1109/TCSVT.2010.2087433

[33] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, L. K. Cormack, Study of subjective and objective quality assessment of video, Trans. Img. Proc. 19 (6) (2010) 1427–1441. doi:10.1109/TIP.2010.2042111.

[34] F. J. Suárez, A. García, J. C. Granda, D. F. García, P. Nuño, Assessing the qoe in video services over lossy networks, J. Netw. Syst. Manage. 24 (1) (2016) 116–139. doi:10.1007/s10922-015-9343-y.

[35] M. G. Kendall, A. Stuart, J. K. Ord (Eds.), Kendall's Advanced Theory of Statistics, Oxford University Press, Inc., New York, NY, USA, 1987.

[36] PacketStorm, Packetstorm hurricane ii network emulator, Available at http://packetstorm.com/packetstorm-products/hurricane-ii-software/.

[37] F. Yang, S. Wan, Bitstream-based quality assessment for networked video: a review, IEEE Communications Magazine 50 (11) (2012) 203–209. doi:10.1109/MCOM.2012.6353702.

[38] A. Liotta, D. C. Mocanu, V. Menkovski, L. Cagnetta, G. Exarchakos, Instantaneous video quality assessment for lightweight devices, in: Proc. of Int. Conference on Advances in Mobile Computing, MoMM '13, New York, NY, USA, 2013, pp. 525:525–525:531. doi:10.1145/2536853.2536903.

[39] J. Hu, H. Wildfeuer, Use of content complexity factors in video over ip quality monitoring, in: Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on, 2009, pp. 216–221. doi:10.1109/QOMEX.2009.5246950.

[40] A. Forsythe, Visual Complexity: Is That All There Is?, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 158–166. doi:10.1007/978-3-642-02728-4_17.
URL http://dx.doi.org/10.1007/978-3-642-02728-4_17

[41] M. G. Choi, J. H. Jung, J. W. Jeon, No-reference image quality assessment using blur and noise, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering 3 (2) (2009) 184 – 188.
URL http://waset.org/Publications?p=26

[42] H. R. Wu, M. Yuen, A generalized block-edge impairment metric for video coding, Signal Processing Letters, IEEE 4 (11) (1997) 317–320. doi:10.1109/97.641398.

[43] S. S. Hemami, A. R. Reibman, No-reference image and video quality estimation: Applications and human-motivated design., Signal Processing: Image Communication 25 (7) (2010) 469–481.
URL http://dblp.uni-trier.de/db/journals/spic/spic25.html#HemamiR10

[44] G. Gescheider, Psychophysics: The Fundamentals, Taylor & Francis, 2013. URL https://books.google.es/books?id=gATPDTj8QoYC

[45] D. Mocanu, A. Liotta, A. Ricci, M. Vega, G. Exarchakos, When does lower bitrate give higher quality in modern video services?, in: Network Operations and Management Symposium (NOMS), 2014 IEEE, 2014, pp. 1–5. doi:10.1109/NOMS.2014.6838400.

[46] M. Mohri, A. Rostamizadeh, A. Talwalkar, Foundations of Machine Learning, The MIT Press, 2012.

[47] D. Freedman, Statistical Models : Theory and Practice, Cambridge University Press, 2005. doi:10.1111/j.1751-5823.2010.00122_11.x.

[48] J. R. Quinlan, Learning efficient classification procedures and their application to chess end games, in: Machine Learning. An Artificial Intelligence Approach, 1983, pp. 463–482. doi:10.1007/978-3-662-12405-5_15.

[49] D. W. Opitz, R. Maclin, Popular ensemble methods: An empirical study., J. Artif. Intell. Res. (JAIR) 11 (1999) 169–198. doi:10.1613/jair.614.

[50] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140. doi:10.1023/A:1018054314350.

[51] L. Breiman, Bias, variance, and arcing classifiers, Tech. Rep. 460, Statistics Department, University of California at Berkeley (1996).

[52] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139. doi:10.1006/jcss.1997.1504.

[53] J. H. Friedman, Greedy function approximation: A gradient boosting machine, Annals of Statistics 29 (2000) 1189–1232. doi:10.1214/aos/1013203451.

[54] R. Turner, A model explanation system, in: Black Box Learning and Inference NIPS Workshop 2015, 2015.

[55] G. Matheron, Principles of geostatistics, Vol. 58, Society of Economic Geologists, 1963. doi:10.2113/gsecongeo.58.8.1246.

[56] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297. doi:10.1023/A:1022627411411.

[57] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

[58] Matlab, Statistics and machine learning toolbox, Available at http://nl.mathworks.com/products/statistics/.

[59] Matlab, Neural networks toolbox, Available at http://nl.mathworks.com/products/neural-network/.

[60] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (2011) 27:1–27:27, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[61] V. Menkovski, A. Liotta, Intelligent control for adaptive video streaming, in: Consumer Electronics (ICCE), 2013 IEEE International Conference on, 2013, pp. 127–128. doi:10.1109/ICCE.2013.6486825.

[62] D. C. Mocanu, J. Pokhrel, J. P. Garella, J. Seppänen, E. Liotou, M. Narwaria, No-reference video quality measurement: added value of machine learning, Journal of Electronic Imaging 24 (6) (2015) 061208. doi:10.1117/1.JEI.24.6.061208.

[63] P. Gastaldo, S. Rovetta, R. Zunino, Objective quality assessment of mpeg-2 video streams by using cbp neural networks, Neural Networks, IEEE Transactions on 13 (4) (2002) 939–947. doi:10.1109/TNN.2002.1021894.

[64] P. Le Callet, C. Viard-Gaudin, D. Barba, A convolutional neural network approach for objective video quality assessment, Neural Networks, IEEE Transactions on 17 (5) (2006) 1316–1327. doi:10.1109/TNN.2006.879766.

[65] K. Zhu, C. Li, V. Asari, D. Saupe, No-reference video quality assessment based on artifact measurement and statistical analysis, Circuits and Systems for Video Technology, IEEE Transactions on PP (99) (2014) 1–1. doi:10.1109/TCSVT.2014.2363737.

[66] N. Staelens, D. Deschrijver, E. Vladislavleva, B. Vermeulen, T. Dhaene, P. Demeester, Constructing a no-reference h.264/avc bitstream-based video quality metric using genetic programming-based symbolic regression, Circuits and Systems for Video Technology, IEEE Transactions on 23 (8) (2013) 1322–1333. doi:10.1109/TCSVT.2013.2243052.

[67] J. Sogaard, S. Forchhammer, J. Korhonen, No-reference video quality assessment using codec analysis, Circuits and Systems for Video Technology, IEEE Transactions on PP (99) (2015) 1–1. doi:10.1109/TCSVT.2015.2397207.